JC08 Rec'd PCT/PTO 12 APR 2001

FORM PTO-1390 REV. 5-93 US DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE **TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371**		ATTORNEYS DOCKET NUMBER **P01,0062** U.S.APPLICATION NO. (if known, see 37 CFR 1.5) 09/807352
INTERNATIONAL APPLICATION NO. **PCT/EP99/07632**	INTERNATIONAL FILING DATE 12 October 1999	PRIORITY DATE CLAIMED 12 October 1998

TITLE OF INVENTION
"DATA BUS AND METHOD FOR ESTABLISHING COMMUNICATION BETWEEN TWO MODULES BY MEANS OF SUCH A DATA BUS"

APPLICANT(S) FOR DO/EO/US
Robert BAUMGARTNER and Norbert HERSCH

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay.
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.

5. ☒ A copy of International Application as filed (35 U.S.C. 371(c)(2))
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US)
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2).

7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. §371(c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.

8. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).

9. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).)

10. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C.371(c)(5)).

Items 11. to 16. below concern other document(s) or information included:

11. ☒ An Information Disclosure Statement under 37 C.F.R. 1.97 and 1.98; **(PTO 1449, Prior Art, Search Report).**

12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 C.F.R. 3.28 and 3.31 is included. **(SEE ATTACHED ENVELOPE)**

13. ☒ A FIRST preliminary amendment.
    ☐ A SECOND or SUBSEQUENT preliminary amendment.

14. ☒ A substitute specification & marked up version of application.

15. ☐ A change of power of attorney and/or address letter.

16. ☒ Other items or information:
    a. ☒ Submittal of Drawings
    b. ☒ **EXPRESS MAIL #EL 843728447US, dated April 12, 2001.**

CHI_DOCS2:CS2\425807.1 04.12.01 10.05

U.S. APPLICATION NO. (if known, see 37 C.F.R. 1.5)	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
09/807352	PCT/EP99/07632	P01,0062

17. ☒ The following fees are submitted:	CALCULATIONS	PTO USE ONLY
BASIC NATIONAL FEE (37 C.F.R. 1.492(a)(1)-(5): Search Report has been prepared by the EPO or JPO .... \$860.00 International preliminary examination fee paid to USPTO (37 C.F.R. 1.482) .. \$700.00 No international preliminary examination fee paid to USPTO (37 C.F.R. 1.482) but international search fee paid to USPTO (37 C.F.R. 1.445(a)(2) .... \$770.00 Neither international preliminary examination fee (37 C.F.R. 1.482) nor international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO .... \$1040.00 International preliminary examination fee paid to USPTO (37 C.F.R. 1.482) and all claims satisfied provisions of PCT Article 33(2)-(4) .... \$ 96.00		
ENTER APPROPRIATE BASIC FEE AMOUNT =	\$ 860.00	
Surcharge of \$130.00 for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 C.F.R. 1.492(e)).	\$	

Claims	Number Filed	Number Extra	Rate		
Total Claims	18 - 20 =		X \$ 18.00	\$	
Independent Claims	3 - 3 =		X \$ 80.00	\$	
Multiple Dependent Claims			\$270.00+	\$	

TOTAL OF ABOVE CALCULATIONS =	\$ 860.00	
Reduction by 1/2 for filing by small entity, if applicable. Verified Small Entity statement must also be filed. (Note 37 C.F.R. 1.9, 1.27, 1.28)	\$	
SUBTOTAL =	\$ 860.00	
Processing fee of \$130.00 for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492(f)). +	\$	
TOTAL NATIONAL FEE =	\$ 860.00	
Fee for recording the enclosed assignment (37 C.F.R. 1.21(h). The assignment must be accompanied by an appropriate cover sheet (37 C.F.R. 3.28, 3.31). \$40.00 per property +		
TOTAL FEES ENCLOSED =	\$ 860.00	
	Amount to be refunded	\$
	charged	\$

a. ☒ A check in the amount of \$ 860.00 to cover the above fees is enclosed.

b. ☐ Please charge my Deposit Account No. ________ in the amount of \$ ______ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 501519. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 C.F.R. 1.494 or 1.495 has not been met, a petition to revive (37 C.F.R. 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Schiff Hardin & Waite
Patent Department
6600 Sears Tower
Chicago, Illinois 60606

CUSTOMER NO. 26574

[signature]
SIGNATURE

Melvin A. Robinson
NAME

31,870
Registration Number

IN THE UNITED STATES ELECTED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

"**PRELIMINARY AMENDMENT**"

APPLICANT: Robert BAUMGARTNER et al.

SERIAL NO.: EXAMINER:

FILING DATE: ART UNIT:

INTERNATIONAL APPLICATION NO.: PCT/EP99/07632

INTERNATIONAL FILING DATE: 12 October 1999

INVENTION: DATA BUS AND METHOD FOR ESTABLISHING COMMUNICATION BETWEEN TWO MODULES BY MEANS OF SUCH A DATA BUS

Hon. Assistant Commissioner for Patents
Box PCT
Washington D.C. 20231

SIR:

Amend the above-identified international application before entry into the national stage before the U.S. Patent & Trademark Office under 35 U.S.C. §371 as follows:

IN THE SPECIFICATION

Amend the specification as follows:

S P E C I F I C A T I O N

TITLE

DATABUS AND METHOD FOR THE COMMUNICATION OF TWO ASSEMBLIES BY MEANS OF SUCH A DATABUS

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a parallel databus and to a method for the communication of two assemblies by means of such a databus. The invention particularly refers to a parallel databus, which is suitable for a multiprocessor architecture. Given such a multiprocessor architecture, a plurality of processor systems basically having equal rights can communicate with one another via the databus.

Description of the Related Art

The multibus II (multibus is a registered trademark of the Intel Corporation) represents such a databus. The multibus II is a synchronized bus defined in IEEE Standard for a High-Performance Synchronous 32-bit bus: MULTIBUS II, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, NY 10017, USA, 1988. In order to make it more simple, the "MULTIBUS II" is referred to as "multibus" in the following.

The hardware realization of such a multibus consists of a backplane, in which the signal lines of the bus are arranged and which are provided with approximately 20 cable connectors, whereby an assembly can be respectively

connected thereto. Figure 3 schematically shows two assemblies 2 that are connected via a multibus 1. Each assembly 2 has a databus driver 3 that is immediately connected to the signal lines of the multibus 1 and has a controller 4 that is connected to the databus driver 3. The controller 4, in turn, is connected to the electronic physical units of the assembly 2. These electronic physical units can have a processor or merely represent a passive digital circuit.

The controller 4, corresponding to the protocol of the multibus 1, logically edits the data generated by the electronic physical units and forwards them to the databus driver 3. The databus driver 3 converts the data into electrical data signals that are appropriate for the multibus and applies the electrical data signals to the signal lines. Data signals coming from the multibus 1, in a reversed way, are accepted by the databus driver, which forwards the data to the controller 4. The controller 4 correspondingly edits the data for the processing by the electronic physical units.

The databus drivers are transparent electronic physical units, i.e., the respective corresponding input side and output side of the databus driver assumes the same logical value. Since the databus drivers are transparently fashioned, an active connection between two assemblies 2 is logically through-switched from the controller 4 of the one assembly 2 to the controller 4 of the other assembly 2.

The signal propagation time between the two controllers 4 limits the

maximum transmission frequency or, respectively, bus frequency. Corresponding to the aforementioned IEEE standard, the bus frequency is 10 MHz. A transmission rate of 40 byte/s is obtained by such a bus frequency.

A study "20MHZ MULTIBUS II PARALLEL SYSTEM BUS INVESTIGATION, TAUFIK MA, INTEL CORPORATION, 8 APRIL, 1991" planned to operate the multibus with a bus frequency of 20 MHz. For this purpose, extensive adaptations and modifications have been proposed in order to optimize the individual runtimes between the controllers and databus drivers or, respectively, between the databus drivers connected via the multibus. The aim of this study is to operate a multibus having 10 assemblies at a maximum and 20 MHz, and to operate a multibus having 20 assemblies at a maximum and 16 MHz. The signal propagation time between the controllers of two assemblies would have to be reduced to 50 ns or less. The result of this study is that such an "accelerated" multibus is theoretically possible, however, there would be a considerable developing outlay until its actual realization. Younger data busses, such as the PCI bus, do not have bus drivers in order to obtain faster access to the signal lines of the databus and therefore obtain a higher throughput. These data busses, however, are limited with respect to the number of assemblies to be connected at a maximum, which is normally clearly smaller than 10, and its physical expanse is limited to 10 cm, for example. On the other hand, a multibus

can be up to 50 cm long and can connect 20 assemblies to one another, whereby a plurality of assemblies can represent processor systems of equal rights.

The publication by "Färber, G., Bussysteme, R. Oldenbourg Verlag, Munich 1987 (2. edition)" describes functions and structures of bus systems on the pages 16- 19. On page 19, image 13 shows a handshake transmission. The article "Packer, Stephen et. al., Message Passing Supports Multiple Processor Design, Computer Design" of 15 June 1984, pages 117 - 120, 122 and 124 describes measures for improving the communication in the multibus II.

SUMMARY OF THE INVENTION

An object of the present invention is to create a parallel databus, which allows a high data throughput and which still has the advantages of the known multibus, such as the high number of connectable assemblies, the large physical expanse and the possibility of a multiprocessor architecture. Another object of the present invention is to create a method for the communication between two assemblies, which are respectively provided with a processor, by such a databus.

This object is achieved by a parallel databus having a plurality of parallel signal lines to which a plurality of assemblies can be connected, whereby each assembly has a databus driver being in immediate connection with the signal lines and has a controller that is connected to the databus driver, whereby a sub-number of the signal lines represent data lines for transmitting the data and control lines

for controlling the data transmission of the data via the data lines, and a clock generator for generating a predetermined bus frequency, with which the signals transmitted in the signal lines are clocked, the databus drivers are connected to the clock generator and the databus drivers are fashioned such that the signals to be transmitted from and to the data and control lines are accepted during a clock pulse prescribed by the clock generator, and are emitted during a following clock pulse.

In a preferred embodiment, the clock generator generates a bus frequency of at least 20 MHz. Specifically, the clock generator may generate a bus frequency of approximately 40 MHz. In one embodiment, the databus has 32 data lines. A further sub-number of the signal lines are fashioned as decision lines for deciding which assembly connected to the signal lines has access priority, the decision lines are connected to non-clocked open-drain outputs of the respective databus drivers, so that they form a wired-or logic. A device for generating an auxiliary clock pulse with a lower frequency than the bus frequency is provided for driving the decision lines.

The device for generating an auxiliary clock pulse may be a frequency divider. The outputs of the databus driver leading to the controller may be fashioned as low- voltage TTL outputs. The signal lines preferably have a physical expanse of at least 40 cm. The signal lines may have a physical expanse of at least 50 cm. A

plurality of the assemblies that are connected to the signal lines are respectively can be provided with a processor. The databus may be multibus-compatible.

The invention also provides a method for the communication of two assemblies, which are each connected to a processor, by means of a parallel databus, whereby data packets are exchanged between the two assemblies, each data packet is acknowledged by only one single handshake. In a preferred embodiment, the method a handshake respectively comprises a data-ready signal of the transmitter assembly and a data-ready signal of the receiver assembly, whereby the data-ready signal of the transmitter assembly is sent to the receiver assembly at the beginning of the data transfer, and the receiver assembly sends its data-ready signal to the receiver assembly after the data-ready signal of the transmitter assembly has been received. The transmitter assembly only sends its data-ready signal when the complete data packet is present on this assembly.

The maximum size of the data packets is preferably set to a predetermined value, and the receiver assembly only sends its data-ready signal when there is sufficient storage space on the receiver assembly. A maximum size of 32 byte, 64 byte, 96 byte or 128 byte are determined for the data packets.

The invention also provides a printer control unit for high-performance printers having an I/O-module, one or more raster modules and a serializer module , whereby the modules each have a processor, the modules are connected

to a parallel databus.

The inventive parallel databus has a plurality of parallel signal lines, whereby a plurality of assemblies can be connected thereto, whereby each assembly has a databus driver being in immediate connection with the signal lines and a controller that is connected to the databus driver, whereby a sub-number of the signal lines represent data lines for transmitting the data and control lines for controlling the data transmission of the data via the data lines, and is provided with a clock generator for generating a predetermined bus frequency, with which the signals transmitted in the signal lines are clocked. This parallel databus is characterized in that the databus drivers are connected to the clock generator and in that the databus drivers are fashioned such that the signals to be transmitted from and to the data lines and control lines are accepted during a clock pulse prescribed by a clock generator and are emitted during a subsequent clock pulse. In this way, a signal to be transmitted from one assembly to another assembly, during a first clock pulse, is transmitted from the controller of the transmitter assembly to the databus driver of the transmitter assembly, is transmitted from the databus driver of the transmitter assembly via the signal lines to the databus driver of the receiving assembly during a second clock pulse and is transmitted from the databus driver of the receiving assembly to the controller of the receiving assembly during a third clock pulse. During a clock pulse, the signals are merely

transmitted between a controller and a databus driver of an assembly or between two databus drivers of two different assemblies, so that the physical signal path is kept short. These short signal paths allow correspondingly short signal propagation times, which can be kept less than or equal to 25 ns, for example, so that a bus clock pulse, for example, of 40 MHz is possible. As a result thereof, the data throughput is considerably increased, although all advantages known about the multibus are kept.

The method for the communication of two assemblies, which respectively have a processor, by such a parallel databus is characterized in that the data packets are merely acknowledged by one single handshake when the data packets are exchanged between the two assemblies. Therefore, the handshake can be fashioned so as to be distributed over a number of clock pulses, whereas the data packet can be transmitted with maximum transmission speed (= one data word per clock pulse).

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is subsequently explained on the basis of an exemplary embodiment shown in the drawing.

Figure 1 is a block diagram of a printer control unit with an inventive databus,

Figure 2 is a functional block diagram of two assemblies connected via an

inventive databus,

Figure 3 shows two assemblies connected via a known databus, and

Figure 4 - 12 show timing diagrams for explaining the signal transmission by means of the inventive databus.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The inventive parallel databus 5 is subsequently explained in greater detail on the basis of a printer control unit 6 for high-performance printers 7. Such a printer control unit 6 has an I/O module 8, one or more raster modules 9 and a serializer module 10. The individual modules 8 to 10 are connected to one another via the databus 5. The raster modules 9 and the serializer module 10 are connected to one another via a further pixel bus 11.

The I/O module 8 receives the bits of printing information from a computer means, which can be a large computer system or also a computer network. The I/O module 8 forwards the printing information to the raster modules 9 and the serializer module 10, whereby the raster modules 9 receive the bits of printing format information and convert them into a printing format data stream that can be processed by the high-performance printer 7. These printing format data streams are transmitted by the raster modules 9 via the pixel bus 11 to the serializer module 10, which lines up the data streams in a predetermined sequence and forwards them to the high-performance printer 7.

The modules 8 to 10 represent assemblies 2 that are respectively connected to the databus 5, whereby each assembly has a databus driver 3 and a controller 4 (Figure 2). The databus 5 corresponds to the multibus II (multibus is a registered trademark of Intel Corp.), as it is defined in IEEE standard for "High Performance Synchronous 32-Bit Bus: MULTIBUS II The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, NY 10017, USA, 1998", apart from the changed cited in the following description.

The hardware realization of this databus 5 is composed of a backplane, in which the signal lines of the bus are arranged and which are provided with 20 to 25 cable connectors to which an assembly 2 can be respectively connected. The CSM module (Central Services Module), which executes specific start routines and which initializes the individual assemblies, represents such an assembly known from the multibus. The CSM module has a clock generator, which applies a clock signal oscillating with a predetermined bus frequency to a clock signal line of the databus 5. The bus frequency is 40 MHz in the present exemplary embodiment.

The databus drivers 3 of each assembly 2 are connected to the clock signal line 12, whereby the input and output of the databus drivers 3 can be clocked corresponding to the bus frequency or, respectively, the bus clock pulse.

In addition to the clock signal line 12, the databus 5 has further lines, such

as 32 data lines for transmitting the data, control lines for controlling the data transmission, decision lines for deciding (arbitration), which assembly is allowed to access the databus 5, address lines and one or more lines for the supply voltage and ground. In the present exemplary embodiment, the same lines are used for transmitting the addresses and the data, so that combined address lines/data lines are present.

The databus drivers 3 are inventively connected to the clock signal line 12 and are fashioned such that the signals to be transmitted from and to the data lines and control lines are accepted during a clock pulse prescribed by a clock generator and are outputted during the following clock pulse. The part of the databus driver 3, which operates the data lines and control lines, therefore is fashioned as a non-transparent electronic component with a temporary storing function, as it can be realized by a D-flip-flop, for example. These databus drivers 3, during a clock pulse, therefore accept the signals of the data lines and control lines coming from the databus 5, they store them and output them to the respective controller 4 during the immediately following clock or, respectively, they accept a signal coming from the controller 4 during a clock pulse, they store said signal and apply it to the databus 5 at the immediately following clock pulse. The databus drivers 3 therefore are operated in a "clocked" fashion with respect to the data lines and control lines.

For explanation purposes, it is assumed in the following that the assembly shown in Figure 2 on the left side (transmitter assembly S) initiates a data transfer to the assembly (receiver assembly E) shown on the right side. In a decision method (arbitration) known from the multibus, the transmitter assembly initially obtains the right to be allowed to access the databus 5. The transmitter assembly S therefore is also referred to as bus owner.

During a first clock pulse, the controller 4 of the transmitter assembly S transmits a data word (1 - 4 byte) to the databus driver of the transmitter assembly S. The databus driver 3 stores the data word and converts it into a signal that is suitable for the databus 5, whereby said signal is present at the signal lines of the databus 5 during the following, second clock. During the second clock pulse, these electrical data signals are accepted by the databus driver 3 of the receiver assembly 2, they are temporarily stored and are transmitted to the controller 4 of the receiving assembly E during the following, third clock pulse.

This clocked transmission of the signals divides the entire transmission path from the controller 4 of the transmitter assembly S to the controller 4 of the receiver assembly E into three sections, namely the two sections 13, 15 between the controllers 4 and the databus drivers 3 of the respective assembly 2 and the section 14 between the two databus drivers 3 of the two assemblies 2, whereby said section extends across the signal lines of the databus 5. The signal

propagation time in the individual sections is significantly shorter than over the entire distance between two controllers, as it is necessary for transparent databus drivers in order to maintain the signal generated at the controller of the transmitter assembly until it is present at the controller of the receiver assembly. Given such a databus, the bus frequency can be significantly increased (e.g. up to 40 MHz) as a result of the inventively shortened signal propagation times of the individual sections, which can be 25 ns, for example. Data transmission rates of up to 160 Mbyte/s can be obtained by a bus frequency of 40 MHz.

The databus drivers can be fashioned, for example, with GTL+- drivers of the company Texas Instruments, such as the module SN54GTL1655 or SN74GTL1655.

The operation of the inventive databus is subsequently explained in greater detail on the basis of the timing diagrams shown in the Figures 4 to 12.

Figure 4 shows the timing diagram for transmitting a data packet between two assemblies on the basis of the control signals SCN0, SCN2, SCN3, SNC4 and the address signals/data signals AND. The designations of these signals are respectively provided with one of the following endings "_S", "_B"and "_E", whereby "-S" means that the signal status in the line section 13 is shown between the controller 4 and the databus driver 3 of the transmitter assembly S; the ending "_B" means that the signal status in the signal lines of the databus is shown

between the databus drivers 3 of the communicating assemblies (line section 14), and the ending "_E" means that the signal in the line section 15 is shown between the databus driver 3 and the controller 4 of the receiver assembly E. The signal SCN0 shows a request phase, whereby the signal is valid in the low status (L = request phase). The signal SCN3 indicates the end of the data transmission, whereby the 0/1-transition or, respectively, low/high-transition represents the exact point in time of the end of the data transmission. SCN3 refers to the data-ready signal of the transmitter assembly, and SCN4 refers to the data-ready signal of the receiver assembly E. The two signals SCN3 and SCN4 are the main components of a handshake between the transmitter assembly S and the receiver assembly E. The handshake has signals SCN5, SCN6 and SCN7 allocated, with which the type of error, in a way known from the multibus, is indicated by the receiver assembly E given a faulty data transmission.

The signals AND can comprise 4, 8, 16, 24 or 32 individual signals, for example, whereby the maximum number is limited by the 32 address lines/data lines of the databus 6.

The controller 4 of the transmitter assembly generates the request signal SCN0 at the beginning of a transmission of a data packet and the address of the receiver assembly is outputted (see S1 in Figure 4). The databus driver 3 of the

transmitter assembly applies these signals to the signal lines of the databus 5 during the next clock pulse (S2). The request signal SCN0 and the address data reach the controller 4 of the receiver assembly E during the third clock pulse (S3); the receiver assembly E then realizes that it is to receive a data packet. The signal SCN0 is only generated during the duration of a clock pulse. During the second clock pulse S2, the controller 4 of the transmitter assembly S generates a data-ready signal SCN3, with which it indicates that the transmitter assembly S is ready for sending a data block. An identification character is simultaneously applied to the address lines/data lines for the type of data block. These signals arrive at the controller 4 of the receiver assembly E at the clock pulse S4. After the receiver assembly E has determined the data-ready signal SCN3, it checks whether it has sufficiently free storage space for accepting a data packet. The size of the data packet is fixed at the initialization of the databus and can be 32, 64, 96 or 128 byte. If there is sufficient storage space at the receiver assembly E for accepting a data packet, the controller 4 of the receiver assembly E produces a data-ready signal SCN4, which is transmitted to the controller of the transmitter assembly S during three clock pulses (S6 - S8). The receiver assembly E needs two clock pulses for accepting the data-ready signal SCN3 of the transmitter assembly S, for checking whether there is sufficient storage space and for outputting the data-ready signal SCN4. The data-ready signal SCN4 reaches the

controller 4 of the transmitter assembly S during the clock pulse S8. The handshake has been acknowledged with the receipt of this signal by the controller 4 of the transmitter assembly S.

The controller 4 of the transmitter assembly S sends data words, which normally comprise 32 bit, during the acknowledgment of the handshake, starting with the clock pulse S3 through the clock pulse S10. The controller 4 hereby sends such a data word per clock pulse, which then, with a delay of two clock pulses, arrive at the controller 4 of the receiver assembly E. The data-ready signal SCN3 of the transmitter assembly S is active until the last data word has been sent. At the same time as the last data word D7 is sent, the controller 4 of the transmitter assembly S activates the control signal SCN2 (low) in order to mark the exact end of the transmission of the data packet at its 0/1-transition (low/high transition). As can be seen from Figure 4, the SCN2 signal is synchronously transmitted in the individual transmission sections with the data word D7, so that the signal SCN2, during the clock pulse S12, is received by the receiver assembly E and also by the other assemblies connected to the signal lines of the databus 5, so that these recognize that the transmission of the data packet has been completed. A further data packet now can be transmitted between the transmitter assembly and the receiver assembly or the right to access the databus signals can be transferred to another assembly in a decision method (arbitration), which is

known from the multibus.

Since each data packet, which can comprise 8 data words, for example, is inventively transmitted by one single handshake, the data words D0 - D7 can be transmitted with a maximum transmission rate (one data word per clock pulse), whereas a comparably long period of time of 7 clock pulses (S2 - S8), for example, is available for processing the handshake.

Figure 5 shows a similar diagram for transmitting a data packet that comprises only one data word D0. The acknowledgment of the handshake by the signal SCN4 is received again at the clock pulse S8 by the controller 4 of the transmitter assembly S (see transmission process explained on the basis of Figure 4). Only after the handshake has been acknowledged and this acknowledgment signal SCN4 has been processed by the transmitter assembly S (clock pulse S10), the controller 4 of the transmitter assembly S ends the output of the signals representing the data word D0, so that it is assured that the receiver assembly E is capable of receiving the data word D0. Since only one data word D0 is transmitted here, it is the "last" data word of the packet why the control signal SCN2 indicating the end of the data transmission is also actively switched (low) during the entire time during which the data signals of the data word D0 are active.

The timing diagrams of Figure 6 and 7 show a reference read access and a

reference write access. The reference accesses comprise the what is referred to as "I/O-space operation" and the "Memory Space Operations", wherein a transmitter assembly writes a data word in a register or a memory cell of another assembly or, respectively, reads is out from the memory cell.

Given the reference write access (Figure 6), the process of the control signals SCN0, SCN2, SCN3 and SCN4 exactly corresponds to the process when a data packet having one single data word is transmitted (Figure 5). The reference write access only differs from this data transfer in that an address ADR composed of two data words is initially transmitted at the address lines/data lines and the data word D0 to be transmitted is subsequently transmitted. A transmitter assembly S initiating a data transfer can read out a memory cell of a receiver assembly E with the reference read access (Figure 7). In the same way as the previous data transfers, the data transfer is started by initiating a request phase with the control signal SCN0 and by applying an address ADR, which is composed of two data words, to the address lines/data lines. After the request phase, the transmitter assembly S sets the control signals SCN2 and SCN3, whereby it thus indicates that it is ready for accepting the data from the receiver assembly E. The receiver assembly E sends a data word D0 to the transmitter assembly S and simultaneously indicates that the data are valid in that it places the control signal SCN4. The receiver assembly E removes again the data D0 and the

control signal SCN4 when it has recognized the set signals SCN2 and SCN3 of the transmitter assembly S. As soon as the transmitter assembly S has recognized the set signal SCN4, it takes over the data D0 and resets the control signals SCN2 and SCN3. The data transfer is completed.

The Figures 8 and 9 show the timing diagrams of an interconnect write access and of an interconnect read access. On the basis of these interconnect accesses, a data word can be respectively entered or, respectively, read out in a what is referred to as interconnect storage space, which is provided at each assembly. The memory cells of this storage space are addressed with an address ADR that is only composed of a data word. The interconnect accesses therefore differ from the reference accesses with respect to the size of the address, whereby the operational sequence of the control signals SC0, SCN2, SCN3, SCN4 corresponds to the transmission of the respective data word D0.

The reference access and interconnect access is of secondary importance for the bus systems, which respectively connect assemblies that are provided with a processor, since an interprocessor communication is only possible by means of the above-described data packets (messages). Given multiprocessor systems, the reference access and the interconnect access only serve the purpose of initializing and diagnosing the system or, respectively, are for the communication with periphery devices, which do not have a separate processor control.

As it is known from the multibus, the control signals ARB (5. . 0) and a bus request signal BREQ (bus request) are used for the decision (arbitration). In contrast to the address signals/data signals and the control signals, these signals are not clocked-in, since the decision lines then cannot be used in a "wired-or-modus", which is used for deciding the access rights. The databus driver therefore is transparent for these signals. Since one clock pulse period is not sufficient as signal runtime from one controller to the other controller for transmitting the signals, the databus is provided with an additional clock pulse signal line, whereby an auxiliary clock pulse BCLK2 is applied thereto. The auxiliary clock pulse BCLK2 (20 MHz) is generated by dividing the bus frequency by two.

The signals ARB (5 . . 0) and BREQ are generated by the assemblies in the high-phase of the auxiliary clock pulse and are also queried in the high-phase of the auxiliary clock pulse BCLK2. It is thus assured that at least two clock pulse periods of the bus frequency or, respectively, of the bus clock pulse are available to the signals as signal runtime.

As shown in Figure 10, four clock pulses of the bus clock pulse or, respectively, two clock pulses of the auxiliary clock pulse are used at a minimum for the transiency of the decision signals. This period of time that is available to the decision signal for the transiency can be extended to 18 clock pulses of the bus frequency at a maximum.

Figure 11 shows a diagram showing the termination of a data transmission due to an error generated at the receiver assembly E. When the receiver assembly E notices an error, it outputs an error code at the same time as the data-ready signal SCN4 by means of the control signals SCN4 to SCN7 (not shown). The error code outputted by the signals SCN5 to SCN7 corresponds to the error code known from the multibus.

When the receiver assembly E recognizes the error code, it modifies the data transfer in that it sets the control signal SCN2. The data transfer is completed when the receiver assembly E recognizes the set control signal SCN2.

It is differentiated between what are referred to as bus errors, which generally occur in the databus system, vis-a-vis the errors generated with respect to the assemblies. Given the inventive databus, the error signal BUSERR can be only set by the assemblies, which are a part of the respective data transfer, i.e., the transmitter assembly S and the receiver assembly E. All other assemblies connected to the databus are either in a resting state or are in the decision phase (arbitration). They must leave the arbitration phase and start again when a bus error occurs. The assemblies set the error signal BUSERR when a parity error is present at the address/data signals or control signals or when the bus protocol has been injured with respect to the control signals. This is to assure that the data arriving at an assembly are correct. Given a data transmission by means of a data

packet, a writing interconnect access and a writing reference access, the error signal BUSERR is generated by the receiver assembly. Given a reading interconnect access and a reading reference access, wherein a data word is transmitted from the receiver assembly to the transmitter assembly, the error signal BUSERR is generated by the transmitter assembly.

It is important about the inventive databus that all signals of a transmitter assembly are clocked-in three times before they can be processed by the receiver assembly. This is the reason why a bus error is always recognized two or three clock pulses later, depending on whether the error has occurred on the transmitter assembly or on the databus. In this case, the receiver assembly sets the error signal BUSERR. Given a reading interconnect access or reading reference access, the error occurs on the receiver assembly, whereby the transmitter assembly sets the error signal BUSERR in the case of an error.

The chronological delay of the error signal is taken into consideration in that the controller of the transmitter assembly, which sends a data packet or carries out an access, waits for six clock pulses after the end of the data transfer until it informs the CPU of the assembly of the status of the data transfer, since an error signal can still arrive at this time.

Since only one single handshake inventively occurs for each data transfer, only this one single handshake, i.e. the data-ready signal SCN4, can be checked,

so that modifications result vis-a-vis the known multibus protocol.

The parity check is carried out at the subsequently described points in time:

1. Control signals are always checked.
2. Address signals/data signals are checked during the request phase.
3. Address signals/data signals are checked during a response phase of a read access following the request phase when SCN4 is set and SCN (7 . . 5) is not set, i.e., when a correct handshake is present.
4. Address signals/data signals are checked during the response phase of a write access when SCN3 is set and when SCN (7 . .5) is not set, i.e., a correct handshake is present.

A databus protocol injury is determined when

1. SCN0 is set during a response phase,
2. SCN2 is set in the response phase and SCN3 is not set in the response phase,
3. SCN5, SCN6 or SCN7 are set during the response phase and SCN4 is not set,
4. SCN2 changes during the response phase, whereas SCN3 remains set, and
5. more data words than fixed during the configuration of the databus system are sent during a data transfer by means of data packets.

The multibus protocol treats the error described in point 5 as an error

occurring at an assembly. Since the data transfer can be no longer continued after this error has occurred and since the error normally occurs after the handshake given the inventive method with one single handshake, it is evaluated as a bus error. Such a bus error with respect to a data transfer by a data packet is shown in the diagram of Figure 12. Nine data words are transmitted here instead of eight data words. Therefore, the receiver assembly sets the error signal BUSERR after the eighth data word D7. As can be seen from the timing diagram, the transmitter assembly has already completed the data transfer on its side before it is informed of the error by the receiver assembly. As a result thereof, the controller of the transmitter assembly must wait for six clock pulses until it can announce the completion of the transfer to the CPU of the assembly, since error messages of the receiver assembly can still arrive within this period of time.

The invention is not limited to a databus that is similar to the multibus but can be utilized for each parallel databus.

The invention can be summarized as follows:

It relates to a parallel databus having a plurality of parallel signal lines to which a plurality of assemblies can be connected, whereby each assembly has a databus driver being in immediate connection with the signal lines and has a controller that is connected to the databus driver. This databus is based on the known MULTIBUS II. The invention is characterized in that the databus drivers

are connected to the clock generator of the databus, and that the databus drivers are fashioned such that the signals to be transmitted from and to the data and control lines are accepted during a clock pulse that is predetermined by the clock generator, and are emitted during the following clock pulse. As a result thereof, the signaling path between two assemblies connected via the databus is interrupted at the databus drivers, so that the signals cover a shorter path section during a clock pulse of the databus compared to conventional databuses with transparent databus drivers. The individual signal propagation times are thus reduced, so that the bus frequency of the databus and therefore the data throughput can be significantly increased.

The invention also relates to a method for the communication of two assemblies that are each provided with a processor, whereby data packets are exchanged between the assemblies. The inventive method is characterized in that each data packet is acknowledged by only one single handshake. As a result of this method, data can be transmitted with a maximum transmission rate at the databus.

Although other modifications and changes may be suggested by those skilled in the art, it is the intention of the inventors to embody within the patent warranted hereon all changes and modifications as reasonably and properly come within the scope of their contribution to the art.

IN THE CLAIMS

Amend the claims as follows:

We claim:

1.(Amended) A parallel databus assembly, comprising:

a plurality of parallel signal lines;

a plurality of assemblies connected to said plurality of parallel signal lines, each of said assemblies having

a databus driver being in immediate connection with said signal lines, and

a controller connected to said databus driver,

at least some of said plurality of parallel signal lines being at least one of data lines for transmitting data and control lines for controlling data transmission of the data via said data lines,

a clock generator generating a predetermined bus frequency with which signals transmitted in said signal lines are clocked,

said databus drivers being connected to said clock generator, said databus drivers being fashioned such that signals to be transmitted from and to said data lines and said control lines are accepted during a clock pulse prescribed by said clock generator and are emitted during a following clock pulse.

2. (Amended) A parallel databus assembly according to claim 1, wherein said

clock generator generates a bus frequency of at least 20 MHz.

3. (Amended) A parallel databus assembly according to claim 2, wherein said

clock generator generates a bus frequency of approximately 40 MHz

4. (Amended) A parallel databus assembly according to claim 1, wherein said databus has 32 data lines.

5. (Amended) A parallel databus assembly according to claim 1, wherein further ones of said plurality of parallel signal lines are fashioned as decision lines for deciding which of said plurality of assemblies connected to said parallel signal lines has access priority, and

said databus drivers having non-clocked open-drain outputs connected to said decision lines a wired-or logic is formed.

6. (Amended) A parallel databus assembly according to claim 5, further comprising:

a device for generating an auxiliary clock pulse with a lower frequency than the bus frequency is provided for driving the decision lines.

7. (Amended) A parallel databus assembly according to claim 6, wherein said device for generating an auxiliary clock pulse is a frequency divider.

8. (Amended) A parallel databus assembly according to claim 1, wherein outputs of the databus driver leading to the controller are fashioned as low-voltage TTL outputs.

9. (Amended) A parallel databus assembly according to claim 1, wherein said signal lines have a physical expanse of at least 40 cm.

10. (Amended) A parallel databus assembly according to claim 1, wherein said signal lines have a physical expanse of at least 50 cm.

11. (Amended) A parallel databus assembly according to claim 1, further comprising:

a processor for a plurality of the assemblies that are connected to the signal lines .

12. (Amended) A parallel databus assembly according to claim 1, wherein said databus is multibus-compatible.

13. (Amended) A method for communication of two assemblies which are each connected to a processor by a parallel databus, comprising the steps of: exchanging data packets between the two assemblies; and acknowledging each data packet by only one single handshake.

14. (Amended) A method according to claim 13, wherein said handshake includes a data-ready signal of the transmitter assembly and a data-ready signal of the receiver assembly, the data-ready signal of the transmitter assembly being sent to the receiver assembly at a beginning of the data transfer, and the receiver assembly sending a data-ready signal to the receiver assembly after the data-ready signal of the transmitter assembly has been received.

15. (Amended) A method according to claim 14, wherein said transmitter assembly only sends its data-ready signal when a complete data packet is present on said assembly.

16. (Amended) A method according to claim 14, further comprising the

step of:

setting a maximum size of the data packets to a predetermined value, and

only sending a data-ready signal from the receiver assembly when there is sufficient storage space on the receiver assembly.

17. (Amended) A method according to claim 16, wherein said step of determining a maximum size determines one of 32 bytes and 64 bytes and 96 bytes and 128 bytes as the maximum size of the data packets.

18.(Amended) A printer control unit for high-performance printers, comprising:

an I/O-module,

at least one raster modules and

a serializer module,

a processor for each of said modules, and

a parallel databus.

IN THE ABSTRACT

ABSTRACT OF THE DISCLOSURE

A parallel databus assembly and method includes modules connected to parallel signal lines, each of the modules having a databus driver being in immediate connection with the signal lines and a controller connected to the databus driver. At least some of the parallel signal lines are data lines for transmitting data or control lines for controlling the data transmission of the data via the data lines. A clock generator for generating a bus frequency with which the signals transmitted in the signal lines are clocked is connected to the databus drivers. The databus drivers are fashioned such that the signals to be transmitted from and to the data lines and the control lines are accepted during a clock pulse prescribed by the clock generator and are emitted during a following clock pulse.

REMARKS

The foregoing amendments to the specification and claims under Article 41 of the Patent Cooperation Treaty place the application into a form for prosecution before the U.S. Patent and Trademark Office under 35 U.S.C. §371. Accordingly, entry of these amendments before examination on the merits is hereby requested.

Respectfully submitted,

Melvin A. Robinson (reg. no. 31,870)
Patent Department
Schiff Hardin & Waite
6600 Sears Tower
Chicago, Illinois 60606
Telephone: 312-258-5785

ATTORNEY FOR APPLICANT

VERSION MARKED TO SHOW CHANGES

The specification has been amended as follows:

SPECIFICATION

TITLE

DATABUS AND METHOD FOR THE COMMUNICATION OF TWO ASSEMBLIES BY MEANS OF SUCH A DATABUS

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a parallel databus and to a method for the communication of two assemblies by means of such a databus. The invention particularly refers to a parallel databus, which is suitable for a multiprocessor architecture. Given such a multiprocessor architecture, a plurality of processor systems basically having equal rights can communicate with one another via the databus.

Description of the Related Art

The multibus II (multibus is a registered trademark of the Intel Corporation) represents such a databus. The multibus II is a synchronized bus defined in IEEE Standard for a High-Performance Synchronous 32-bit bus: MULTIBUS II, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, NY 10017, USA, 1988. In order to make it more simple, the "MULTIBUS II" is referred to as "multibus" in the following.

The hardware realization of such a multibus consists of a backplane

[~~backplan [sic]~~], in which the signal lines of the bus are arranged and which are provided with approximately 20 cable connectors, whereby an assembly can be respectively connected thereto. Figure 3 schematically shows two assemblies 2 that are connected via a multibus 1. Each assembly 2 has a databus driver 3 that is immediately connected to the signal lines of the multibus 1 and has a controller 4 that is connected to the databus driver 3. The controller 4, in turn, is connected to the electronic physical units of the assembly 2. These electronic physical units can have a processor or merely represent a passive digital circuit.

The controller 4, corresponding to the protocol of the multibus 1, logically edits the data generated by the electronic physical units and forwards them to the databus driver 3. The databus driver 3 converts the data into electrical data signals that are appropriate for the multibus and applies the electrical data signals to the signal lines. Data signals coming from the multibus 1, in a reversed way, are accepted by the databus driver, which forwards the data to the controller 4. The controller 4 correspondingly edits the data for the processing by the electronic physical units.

The databus drivers are transparent electronic physical units, i.e., the respective corresponding input side and output side of the databus driver assumes the same logical value. Since the databus drivers are transparently fashioned, an active connection between two assemblies 2 is logically through-switched from

the controller 4 of the one assembly 2 to the controller 4 of the other assembly 2.

The signal propagation time between the two controllers 4 limits the maximum transmission frequency or, respectively, bus frequency. Corresponding to the aforementioned IEEE standard, the bus frequency is 10 MHz. A transmission rate of 40 byte/s is obtained by such a bus frequency.

A study "20MHZ MULTIBUS II PARALLEL SYSTEM BUS INVESTIGATION, TAUFIK MA, INTEL CORPORATION, 8 APRIL, 1991" planned to operate the multibus with a bus frequency of 20 MHz. For this purpose, extensive adaptations and modifications have been proposed in order to optimize the individual runtimes between the controllers and databus drivers or, respectively, between the databus drivers connected via the multibus. The aim of this study is to operate a multibus having 10 assemblies at a maximum and 20 MHz, and to operate a multibus having 20 assemblies at a maximum and 16 MHz. The signal propagation time between the controllers of two assemblies would have to be reduced to 50 ns or less. The result of this study is that such an "accelerated" multibus is theoretically possible, however, there would be a considerable developing outlay until its actual realization. Younger data busses, such as the PCI bus, do not have bus drivers in order to obtain faster access to the signal lines of the databus and therefore obtain a higher throughput. These data busses, however, are limited with respect to the number of assemblies to be

connected at a maximum, which is normally clearly smaller than 10, and its physical expanse is limited to 10 cm, for example. On the other hand, a multibus can be up to 50 cm long and can connect 20 assemblies to one another, whereby a plurality of assemblies can represent processor systems of equal rights.

The publication by "Färber, G., Bussysteme, R. Oldenbourg Verlag, Munich 1987 (2. edition)" describes functions and structures of bus systems on the pages 16- 19. On page 19, image 13 shows a handshake transmission. The article "Packer, Stephen et. al., Message Passing Supports Multiple Processor Design, Computer Design" of 15 June 1984, pages 117 - 120, 122 and 124 describes measures for improving the communication in the multibus II.

SUMMARY OF THE INVENTION

An object of the present invention is to create a parallel databus, which allows a high data throughput and which still has the advantages of the known multibus, such as the high number of connectable assemblies, the large physical expanse and the possibility of a multiprocessor architecture. Another object of the present invention is to create a method for the communication between two assemblies, which are respectively provided with a processor, by such a databus.

This object is achieved by a parallel databus having [~~the features of claim 1 and by a method having the features of claim 13. Advantageous embodiments of the invention are cited in the subclaims.~~] a plurality of parallel signal lines to

which a plurality of assemblies can be connected, whereby each assembly has a databus driver being in immediate connection with the signal lines and has a controller that is connected to the databus driver, whereby a sub-number of the signal lines represent data lines for transmitting the data and control lines for controlling the data transmission of the data via the data lines, and a clock generator for generating a predetermined bus frequency, with which the signals transmitted in the signal lines are clocked, the databus drivers are connected to the clock generator and the databus drivers are fashioned such that the signals to be transmitted from and to the data and control lines are accepted during a clock pulse prescribed by the clock generator, and are emitted during a following clock pulse.

In a preferred embodiment, the clock generator generates a bus frequency of at least 20 MHz. Specifically, the clock generator may generate a bus frequency of approximately 40 MHz. In one embodiment, the databus has 32 data lines. A further sub-number of the signal lines are fashioned as decision lines for deciding which assembly connected to the signal lines has access priority, the decision lines are connected to non-clocked open-drain outputs of the respective databus drivers, so that they form a wired-or logic. A device for generating an auxiliary clock pulse with a lower frequency than the bus frequency is provided for driving the decision lines.

The device for generating an auxiliary clock pulse may be a frequency divider. The outputs of the databus driver leading to the controller may be fashioned as low- voltage TTL outputs. The signal lines preferably have a physical expanse of at least 40 cm. The signal lines may have a physical expanse of at least 50 cm. A plurality of the assemblies that are connected to the signal lines are respectively can be provided with a processor. The databus may be multibus-compatible.

The invention also provides a method for the communication of two assemblies, which are each connected to a processor, by means of a parallel databus, whereby data packets are exchanged between the two assemblies, each data packet is acknowledged by only one single handshake. In a preferred embodiment, the method a handshake respectively comprises a data-ready signal of the transmitter assembly and a data-ready signal of the receiver assembly, whereby the data-ready signal of the transmitter assembly is sent to the receiver assembly at the beginning of the data transfer, and the receiver assembly sends its data-ready signal to the receiver assembly after the data-ready signal of the transmitter assembly has been received. The transmitter assembly only sends its data-ready signal when the complete data packet is present on this assembly.

The maximum size of the data packets is preferably set to a predetermined value, and the receiver assembly only sends its data-ready signal when there is sufficient storage space on the receiver assembly. A maximum size of 32 byte, 64

byte, 96 byte or 128 byte are determined for the data packets.

The invention also provides a printer control unit for high-performance printers having an I/O-module, one or more raster modules and a serializer module , whereby the modules each have a processor, the modules are connected to a parallel databus.

The inventive parallel databus has a plurality of parallel signal lines, whereby a plurality of assemblies can be connected thereto, whereby each assembly has a databus driver being in immediate connection with the signal lines and a controller that is connected to the databus driver, whereby a sub-number of the signal lines represent data lines for transmitting the data and control lines for controlling the data transmission of the data via the data lines, and is provided with a clock generator for generating a predetermined bus frequency, with which the signals transmitted in the signal lines are clocked. This parallel databus is characterized in that the databus drivers are connected to the clock generator and in that the databus drivers are fashioned such that the signals to be transmitted from and to the data lines and control lines are accepted during a clock pulse prescribed by a clock generator and are emitted during a subsequent clock pulse. In this way, a signal to be transmitted from one assembly to another assembly, during a first clock pulse, is transmitted from the controller of the transmitter assembly to the databus driver of the transmitter assembly, is transmitted from the

databus driver of the transmitter assembly via the signal lines to the databus driver of the receiving assembly during a second clock pulse and is transmitted from the databus driver of the receiving assembly to the controller of the receiving assembly during a third clock pulse. During a clock pulse, the signals are merely transmitted between a controller and a databus driver of an assembly or between two databus drivers of two different assemblies, so that the physical signal path is kept short. These short signal paths allow correspondingly short signal propagation times, which can be kept less than or equal to 25 ns, for example, so that a bus clock pulse, for example, of 40 MHz is possible. As a result thereof, the data throughput is considerably increased, although all advantages known about the multibus are kept.

The method [~~cited in claim 13~~] for the communication of two assemblies, which respectively have a processor, by such a parallel databus is characterized in that the data packets are merely acknowledged by one single handshake when the data packets are exchanged between the two assemblies. Therefore, the handshake can be fashioned so as to be distributed over a number of clock pulses, whereas the data packet can be transmitted with maximum transmission speed (= one data word per clock pulse).

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is subsequently explained on the basis of an exemplary

embodiment shown in the drawing. [Schematically shown are:]

Figure 1 is a block diagram of a printer control unit with an inventive databus,

Figure 2 is a functional block diagram of two assemblies connected via an inventive databus,

Figure 3 shows two assemblies connected via a known databus, and

Figure 4 - 12 show timing diagrams for explaining the signal transmission by means of the inventive databus.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The inventive parallel databus 5 is subsequently explained in greater detail on the basis of a printer control unit 6 for high-performance printers 7. Such a printer control unit 6 has an I/O module 8, one or more raster modules 9 and a serializer module 10. The individual modules 8 to 10 are connected to one another via the databus 5. The raster modules 9 and the serializer module 10 are connected to one another via a further pixel bus 11.

The I/O module 8 receives the bits of printing information from a computer means, which can be a large computer system or also a computer network. The I/O module 8 forwards the printing information to the raster modules 9 and the serializer module 10, whereby the raster modules 9 receive the bits of printing format information and convert them into a printing format data

stream that can be processed by the high-performance printer 7. These printing format data streams are transmitted by the raster modules 9 via the pixel bus 11 to the serializer module 10, which lines up the data streams in a predetermined sequence and forwards them to the high-performance printer 7.

The modules 8 to 10 represent assemblies 2 that are respectively connected to the databus 5, whereby each assembly has a databus driver 3 and a controller 4 (Figure 2). The databus 5 corresponds to the multibus II (multibus is a registered trademark of Intel Corp.), as it is defined in IEEE standard for "High Performance Synchronous 32-Bit Bus: MULTIBUS II The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, NY 10017, USA, 1998", apart from the changed cited in the following description.

The hardware realization of this databus 5 is composed of a backplane [~~backplan [sic]~~], in which the signal lines of the bus are arranged and which are provided with 20 to 25 cable connectors to which an assembly 2 can be respectively connected. The CSM module (Central Services Module), which executes specific start routines and which initializes the individual assemblies, represents such an assembly known from the multibus. The CSM module has a clock generator, which applies a clock signal oscillating with a predetermined bus frequency to a clock signal line of the databus 5. The bus frequency is 40 MHz in the present exemplary embodiment.

The databus drivers 3 of each assembly 2 are connected to the clock signal line 12, whereby the input and output of the databus drivers 3 can be clocked corresponding to the bus frequency or, respectively, the bus clock pulse.

In addition to the clock signal line 12, the databus 5 has further lines, such as 32 data lines for transmitting the data, control lines for controlling the data transmission, decision lines for deciding (arbitration), which [~~whichen [sic]~~] assembly is allowed to access the databus 5, address lines and one or more lines for the supply voltage and ground. In the present exemplary embodiment, the same lines are used for transmitting the addresses and the data, so that combined address lines/data lines are present.

The databus drivers 3 are inventively connected to the clock signal line 12 and are fashioned such that the signals to be transmitted from and to the data lines and control lines are accepted during a clock pulse prescribed by a clock generator and are outputted during the following clock pulse. The part of the databus driver 3, which operates the data lines and control lines, therefore is fashioned as a non-transparent electronic component with a temporary storing function, as it can be realized by a D-flip-flop, for example. These databus drivers 3, during a clock pulse, therefore accept the signals of the data lines and control lines coming from the databus 5, they store them and output them to the respective controller 4 during the immediately following clock or, respectively, they accept a signal

coming from the controller 4 during a clock pulse, they store said signal and apply it to the databus 5 at the immediately following clock pulse. The databus drivers 3 therefore are operated in a "clocked" fashion with respect to the data lines and control lines.

For explanation purposes, it is assumed in the following that the assembly shown in Figure 2 on the left side (transmitter assembly S) initiates a data transfer to the assembly (receiver assembly E) shown on the right side. In a decision method (arbitration) known from the multibus, the transmitter assembly initially obtains the right to be allowed to access the databus 5. The transmitter assembly S therefore is also referred to as bus owner.

During a first clock pulse, the controller 4 of the transmitter assembly S transmits a data word (1 - 4 byte) to the databus driver of the transmitter assembly S. The databus driver 3 stores the data word and converts it into a signal that is suitable for the databus 5, whereby said signal is present at the signal lines of the databus 5 during the following, second clock. During the second clock pulse, these electrical data signals are accepted by the databus driver 3 of the receiver assembly 2, they are temporarily stored and are transmitted to the controller 4 of the receiving assembly E during the following, third clock pulse.

This clocked transmission of the signals divides the entire transmission path from the controller 4 of the transmitter assembly S to the controller 4 of the

receiver assembly E into three sections, namely the two sections 13, 15 between the controllers 4 and the databus drivers 3 of the respective assembly 2 and the section 14 between the two databus drivers 3 of the two assemblies 2, whereby said section extends across the signal lines of the databus 5. The signal propagation time in the individual sections is significantly shorter than over the entire distance between two controllers, as it is necessary for transparent databus drivers in order to maintain the signal generated at the controller of the transmitter assembly until it is present at the controller of the receiver assembly. Given such a databus, the bus frequency can be significantly increased (e.g. up to 40 MHz) as a result of the inventively shortened signal propagation times of the individual sections [~~sectionst [sic]~~], which can be 25 ns, for example. Data transmission rates of up to 160 Mbyte/s can be obtained by a bus frequency of 40 MHz.

The databus drivers can be fashioned, for example, with GTL+- drivers of the company Texas Instruments, such as the module SN54GTL1655 or SN74GTL1655.

The operation of the inventive databus is subsequently explained in greater detail on the basis of the timing diagrams shown in the Figures 4 to 12.

Figure 4 shows the timing diagram for transmitting a data packet between two assemblies on the basis of the control signals SCN0, SCN2, SCN3, SNC4 and the address signals/data signals AND. The designations of these signals are

respectively provided with one of the following endings "_S", "_B"and "_E", whereby "-S" means that the signal status in the line section 13 is shown between the controller 4 and the databus driver 3 of the transmitter assembly S; the ending "_B" means that the signal status in the signal lines of the databus is shown between the databus drivers 3 of the communicating assemblies (line section 14), and the ending "_E" means that the signal in the line section 15 is shown between the databus driver 3 and the controller 4 of the receiver assembly E. The signal SCN0 shows a request phase, whereby the signal is valid in the low status (L = request phase). The signal SCN3 indicates the end of the data transmission, whereby the 0/1-transition or, respectively, low/high-transition represents the exact point in time of the end of the data transmission. SCN3 refers to the data-ready signal of the transmitter assembly, and SCN4 refers to the data-ready signal of the receiver assembly E. The two signals SCN3 and SCN4 are the main components of a handshake between the transmitter assembly S and the receiver assembly E. The handshake has signals SCN5, SCN6 and SCN7 allocated, with which the type of error, in a way known from the multibus, is indicated by the receiver assembly E given a faulty data transmission.

The signals AND can comprise 4, 8, 16, 24 or 32 individual signals, for example, whereby the maximum number is limited by the 32 address lines/data

lines of the databus 6.

The controller 4 of the transmitter assembly generates the request signal SCN0 at the beginning of a transmission of a data packet and the address of the receiver assembly is outputted (see S1 in Figure 4). The databus driver 3 of the transmitter assembly applies these signals to the signal lines of the databus 5 during the next clock pulse (S2). The request signal SCN0 and the address data reach the controller 4 of the receiver assembly E during the third clock pulse (S3); the receiver assembly E then realizes that it is to receive a data packet. The signal SCN0 is only generated during the duration of a clock pulse. During the second clock pulse S2, the controller 4 of the transmitter assembly S generates a data-ready signal SCN3, with which it indicates that the transmitter assembly S is ready for sending a data block. An identification character is simultaneously applied to the address lines/data lines for the type of data block. These signals arrive at the controller 4 of the receiver assembly E at the clock pulse S4. After the receiver assembly E has determined the data-ready signal SCN3, it checks whether it has sufficiently free storage space for accepting a data packet. The size of the data packet is fixed at the initialization of the databus and can be 32, 64, 96 or 128 byte. If there is sufficient storage space at the receiver assembly E for accepting a data packet, the controller 4 of the receiver assembly E produces a data-ready signal SCN4, which is transmitted to the controller of the transmitter

assembly S during three clock pulses (S6 - S8). The receiver assembly E needs two clock pulses for accepting the data-ready signal SCN3 of the transmitter assembly S, for checking whether there is sufficient storage space and for outputting the data-ready signal SCN4. The data-ready signal SCN4 reaches the controller 4 of the transmitter assembly S during the clock pulse S8. The handshake has been acknowledged with the receipt of this signal by the controller 4 of the transmitter assembly S.

The controller 4 of the transmitter assembly S sends data words, which normally comprise 32 bit, during the acknowledgment of the handshake, starting with the clock pulse S3 through the clock pulse S10. The controller 4 hereby sends such a data word per clock pulse, which then, with a delay of two clock pulses, arrive [~~sic~~] at the controller 4 of the receiver assembly E. The data-ready signal SCN3 of the transmitter assembly S is active until the last data word has been sent. At the same time as the last data word D7 is sent, the controller 4 of the transmitter assembly S activates the control signal SCN2 (low) in order to mark the exact end of the transmission of the data packet at its 0/1-transition (low/high transition). As can be seen from Figure 4, the SCN2 signal is synchronously transmitted in the individual transmission sections with the data word D7, so that the signal SCN2, during the clock pulse S12, is received by the receiver assembly E and also by the other assemblies connected to the signal lines

of the databus 5, so that these recognize that the transmission of the data packet has been completed. A further data packet now can be transmitted between the transmitter assembly and the receiver assembly or the right to access the databus signals [[~~sic~~]] can be transferred to another assembly in a decision method (arbitration), which is known from the multibus.

Since each data packet, which can comprise 8 data words, for example, is inventively transmitted by one single handshake, the data words D0 - D7 can be transmitted with a maximum transmission rate (one data word per clock pulse), whereas a comparably long period of time of 7 clock pulses (S2 - S8), for example, is available for processing the handshake.

Figure 5 shows a similar diagram for transmitting a data packet that comprises only one data word D0. The acknowledgment of the handshake by the signal SCN4 is received again at the clock pulse S8 by the controller 4 of the transmitter assembly S (see transmission process explained on the basis of Figure 4). Only after the handshake has been acknowledged and this acknowledgment signal SCN4 has been processed by the transmitter assembly S (clock pulse S10), the controller 4 of the transmitter assembly S ends the output of the signals representing the data word D0, so that it is assured that the receiver assembly E is capable of receiving the data word D0. Since only one data word D0 is transmitted here, it is the "last" data word of the packet why the control signal

SCN2 indicating the end of the data transmission is also actively switched (low) during the entire time during which the data signals of the data word D0 are active.

The timing diagrams of Figure 6 and 7 show a reference read access and a reference write access. The reference accesses comprise the what is referred to as "I/O-space operation" and the "Memory Space Operations", wherein a transmitter assembly writes a data word in a register or a memory cell of another assembly or, respectively, reads is out from the memory cell.

Given the reference write access (Figure 6), the process of the control signals SCN0, SCN2, SCN3 and SCN4 exactly corresponds to the process when a data packet having one single data word is transmitted (Figure 5). The reference write access only differs from this data transfer in that an address ADR composed of two data words is initially transmitted at the address lines/data lines and the data word D0 to be transmitted is subsequently transmitted. A transmitter assembly S initiating a data transfer can read out a memory cell of a receiver assembly E with the reference read access (Figure 7). In the same way as the previous data transfers, the data transfer is started by initiating a request phase with the control signal SCN0 and by applying an address ADR, which is composed of two data words, to the address lines/data lines. After the request phase, the transmitter assembly S sets the control signals SCN2 and SCN3,

whereby it thus indicates that it is ready for accepting the data from the receiver assembly E. The receiver assembly E sends a data word D0 to the transmitter assembly S and simultaneously indicates that the data are valid in that it places the control signal SCN4. The receiver assembly E removes again the data D0 and the control signal SCN4 when it has recognized the set signals SCN2 and SCN3 of the transmitter assembly S. As soon as the transmitter assembly S has recognized the set signal SCN4, it takes over the data D0 and resets the control signals SCN2 and SCN3. The data transfer is completed.

The Figures 8 and 9 show the timing diagrams of an interconnect write access and of an interconnect read access. On the basis of these interconnect accesses, a data word can be respectively entered or, respectively, read out in a what is referred to as interconnect storage space, which is provided at each assembly. The memory cells of this storage space are addressed with an address ADR that is only composed of a data word. The interconnect accesses therefore differ from the reference accesses with respect to the size of the address, whereby the operational sequence of the control signals SC0, SCN2, SCN3, SCN4 corresponds to the transmission of the respective data word D0.

The reference access and interconnect access is of secondary importance for the bus systems, which respectively connect assemblies that are provided with a processor, since an interprocessor communication is only possible by means of

the above-described data packets (messages). Given multiprocessor systems, the reference access and the interconnect access only serve the purpose of initializing and diagnosing the system or, respectively, are for the communication with periphery devices, which do not have a separate processor control.

As it is known from the multibus, the control signals ARB (5. . 0) and a bus request signal BREQ (bus request) are used for the decision (arbitration). In contrast to the address signals/data signals and the control signals, these signals are not clocked-in, since the decision lines then cannot be used in a "wired-or-modus", which is used for deciding the access rights. The databus driver therefore is transparent for these signals. Since one clock pulse period is not sufficient as signal runtime from one controller to the other controller for transmitting the signals, the databus is provided with an additional clock pulse signal line, whereby an auxiliary clock pulse BCLK2 is applied thereto. The auxiliary clock pulse BCLK2 (20 MHz) is generated by dividing the bus frequency by two.

The signals ARB (5 . . 0) and BREQ are generated by the assemblies in the high-phase of the auxiliary clock pulse and are also queried in the high-phase of the auxiliary clock pulse BCLK2. It is thus assured that at least two clock pulse periods of the bus frequency or, respectively, of the bus clock pulse are available to the signals as signal runtime.

As shown in Figure 10, four clock pulses of the bus clock pulse or,

respectively, two clock pulses of the auxiliary clock pulse are used at a minimum for the transiency of the decision signals. This period of time that is available to the decision signal for the transiency can be extended to 18 clock pulses of the bus frequency at a maximum.

Figure 11 shows a diagram showing the termination of a data transmission due to an error generated at the receiver assembly E. When the receiver assembly E notices an error, it outputs an error code at the same time as the data-ready signal SCN4 by means of the control signals SCN4 to SCN7 (not shown). The error code outputted by the signals SCN5 to SCN7 corresponds to the error code known from the multibus.

When the receiver assembly E recognizes the error code, it modifies the data transfer in that it sets the control signal SCN2. The data transfer is completed when the receiver assembly E recognizes the set control signal SCN2.

It is differentiated between what are referred to as bus errors, which generally occur in the databus system, vis-a-vis [[~~sic~~]] the errors generated with respect to the assemblies. Given the inventive databus, the error signal BUSERR can be only set by the assemblies, which are a part of the respective data transfer, i.e., the transmitter assembly S and the receiver assembly E. All other assemblies connected to the databus are either in a resting state or are in the decision phase (arbitration). They must leave the arbitration phase and start again when a bus

error occurs. The assemblies set the error signal BUSERR when a parity error is present at the address/data signals or control signals or when the bus protocol has been injured with respect to the control signals. This is to assure that the data arriving at an assembly are correct. Given a data transmission by means of a data packet, a writing interconnect access and a writing reference access, the error signal BUSERR is generated by the receiver assembly. Given a reading interconnect access and a reading reference access, wherein a data word is transmitted from the receiver assembly to the transmitter assembly, the error signal BUSERR is generated by the transmitter assembly.

It is important about the inventive databus that all signals of a transmitter assembly are clocked-in three times before they can be processed by the receiver assembly. This is the reason why a bus error is always recognized two or three clock pulses later, depending on whether the error has occurred on the transmitter assembly or on the databus. In this case, the receiver assembly sets the error signal BUSERR. Given a reading interconnect access or reading reference access, the error occurs on the receiver assembly, whereby the transmitter assembly sets the error signal BUSERR in the case of an error.

The chronological delay of the error signal is taken into consideration in that the controller of the transmitter assembly, which sends a data packet or carries out an access, waits for six clock pulses after the end of the data transfer

until it informs the CPU of the assembly of the status of the data transfer, since an error signal can still arrive at this time.

Since only one single handshake inventively occurs for each data transfer, only this one single handshake, i.e. the data-ready signal SCN4, can be checked, so that modifications result vis-a-vis the known multibus protocol.

The parity check is carried out at the subsequently described points in time:

1. Control signals are always checked.
2. Address signals/data signals are checked during the request phase.
3. Address signals/data signals are checked during a response phase of a read access following the request phase when SCN4 is set and SCN (7 . . 5) is not set, i.e., when a correct handshake is present.
4. Address signals/data signals are checked during the response phase of a write access when SCN3 is set and when SCN (7 . .5) is not set, i.e., a correct handshake is present.

A databus protocol injury is determined when

1. SCN0 is set during a response phase,
2. SCN2 is set in the response phase and SCN3 is not set in the response phase,
3. SCN5, SCN6 or SCN7 are set during the response phase and SCN4 is not set,

4. SCN2 changes during the response phase, whereas SCN3 remains set, and

5. more data words than fixed during the configuration of the databus system are sent during a data transfer by means of data packets.

The multibus protocol treats the error described in point 5 as an error occurring at an assembly. Since the data transfer can be no longer continued after this error has occurred and since the error normally occurs after the handshake given the inventive method with one single handshake, it is evaluated as a bus error. Such a bus error with respect to a data transfer by a data packet is shown in the diagram of Figure 12. Nine data words are transmitted here instead of eight data words. Therefore, the receiver assembly sets the error signal BUSERR after the eighth data word D7. As can be seen from the timing diagram, the transmitter assembly has already completed the data transfer on its side before it is informed of the error by the receiver assembly. As a result thereof, the controller of the transmitter assembly must wait for six clock pulses until it can announce the completion of the transfer to the CPU of the assembly, since error messages of the receiver assembly can still arrive within this period of time.

The invention is not limited to a databus that is similar to the multibus but can be utilized for each parallel databus.

The invention can be summarized as follows:

It relates to a parallel databus having a plurality of parallel signal lines to

which a plurality of assemblies can be connected, whereby each assembly has a databus driver being in immediate connection with the signal lines and has a controller that is connected to the databus driver. This databus is based on the known MULTIBUS II. The invention is characterized in that the databus drivers are connected to the clock generator of the databus, and that the databus drivers are fashioned such that the signals to be transmitted from and to the data and control lines are accepted during a clock pulse that is predetermined by the clock generator, and are emitted during the following clock pulse. As a result thereof, the signaling path between two assemblies connected via the databus is interrupted at the databus drivers, so that the signals cover a shorter path section during a clock pulse of the databus compared to conventional databuses with transparent databus drivers. The individual signal propagation times are thus reduced, so that the bus frequency of the databus and therefore the data throughput can be significantly increased.

The invention also relates to a method for the communication of two assemblies that are each provided with a processor, whereby data packets are exchanged between the assemblies. The inventive method is characterized in that each data packet is acknowledged by only one single handshake. As a result of this method, data can be transmitted with a maximum transmission rate at the databus.

Although other modifications and changes may be suggested by those skilled in the art, it is the intention of the inventors to embody within the patent warranted hereon all changes and modifications as reasonably and properly come within the scope of their contribution to the art.

[~~Reference character list~~

~~1 multibus~~

~~2 assembly~~

~~3 databus driver~~

~~4 controller~~

~~5 databus~~

~~6 printer control unit~~

~~7 high-performance printer~~

~~8 I/O-module~~

~~9 raster module~~

~~10 serializer module~~

~~11 pixel bus~~

~~12 clock pulse signal line~~

~~13 line section between the controller and the databus driver in the transmitter assembly~~

~~14 line section between the databus drivers of two communicating assemblies~~

~~15 line section between the controller and the databus driver in the receiver assembly~~]

The claims have been amended as follows:

We claim: [~~Patent claims~~]

1.(Amended) A parallel [~~Parallel~~] databus assembly, comprising: [~~having~~]

a plurality of parallel signal lines; [~~to which~~]

a plurality of assemblies [~~(2) can be~~] connected to said plurality of parallel signal lines, [~~whereby~~] each of said assemblies having [~~assembly (2) has~~]

a databus driver being in immediate connection with said [~~the~~] signal lines, and [has]

a controller [~~(4) that is~~] connected to said [~~the~~] databus driver [(3)],

at least some of said plurality of parallel [~~whereby a sub-number of the~~] signal lines being at least one of [~~represent~~] data lines for transmitting [~~the~~] data and control lines for controlling [~~the~~] data transmission of the data via said [~~the~~] data lines, [~~and~~]

a clock generator [~~for~~] generating a predetermined bus frequency [~~;~~] with which [~~the~~] signals transmitted in said [~~the~~] signal lines are clocked, [~~c h a r a c t e r i z e d i n t h a t the~~]

said databus drivers being [~~(3) are~~] connected to said [~~the~~] clock generator, [and the] said databus drivers being [~~(3) are~~] fashioned such that [~~the~~] signals to be transmitted from and to said [~~the~~] data lines and said control lines are accepted during a clock pulse prescribed by said [~~the~~] clock generator [~~;~~]

and are emitted during a following clock pulse.

2. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, wherein said [~~characterized in that the~~] clock generator generates a bus frequency of at least 20 MHz.

3. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 2, wherein said [~~characterized in that the~~] clock generator generates a bus frequency of approximately 40 MHz

4. (Amended) A parallel [~~Parallel~~] databus assembly according to claim [~~one of the claims~~] 1, wherein said [~~to 3, characterized in that the~~] databus has 32 data lines.

5. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, wherein further ones of said plurality of parallel [~~one of the claims 1 to 4, characterized in that a further sub-number of the~~] signal lines are fashioned as decision lines for deciding which of said plurality of assemblies [assembly]

connected to said parallel [the] signal lines has access priority, and [~~whereby the decision lines are connected to~~]

said databus drivers having non-clocked open-drain outputs connected to said decision lines [~~of the respective databus drivers (3), so that they form~~] a wired-or logic is formed.

6. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 5, further comprising: [~~c h a r a c t e r i z e d i n t h a t~~]

a device for generating an auxiliary clock pulse [~~(BCLK2~~] with a lower frequency than the bus frequency is provided for driving the decision lines.

7. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 6, wherein said [~~c h a r a c t e r i z e d i n t h a t the~~] device for generating an auxiliary clock pulse is a frequency divider.

8. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, wherein [~~one of the claims 1 to 7, c h a r a c t e r i z e d i n t h a t the~~] outputs of the databus driver [~~(3)~~] leading to the controller [~~(4)~~] are fashioned as low-voltage TTL outputs.

9. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, wherein said [~~one of the claims 1 to 8, characterized in that the~~] signal lines have a physical expanse of at least 40 cm.

10. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, wherein said [~~one of the claims 1 to 8, characterized in that the~~] signal lines have a physical expanse of at least 50 cm.

11. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, further comprising: [~~one of the claims 1 to 10, characterized in that~~] a processor for a plurality of the assemblies [(2)] that are connected to the signal lines [~~are respectively provided with a processor~~].

12. (Amended) A parallel [~~Parallel~~] databus assembly according to claim 1, wherein said [~~one of the claims 1 to 11, characterized in that the~~] databus [~~(5)~~] is multibus-compatible.

13. (Amended) A method [~~Method~~] for [~~the~~] communication of two assemblies [~~(2),~~] which are each connected to a processor [~~,~~] by [~~means of~~]a parallel databus, comprising the steps of: [~~(5) according to one of the claims 1 to~~

~~12, whereby~~]

exchanging data packets [~~(messages) are exchanged~~] between the two assemblies;

and [~~(2), characterized in that~~]

acknowledging each data packet [~~is acknowledged~~] by only one single handshake.

14. (Amended) A method [~~Method~~] according to claim 13, wherein said [~~characterized in that a~~] handshake includes [~~respectively comprises~~] a data-ready signal [~~(SCN3)~~] of the transmitter assembly [~~(2)~~] and a data-ready signal [~~(SCN4)~~] of the receiver assembly [~~(2)~~], [~~whereby~~] the data-ready signal [(SCN3)] of the transmitter assembly being [~~(2) is~~] sent to the receiver assembly [(2)] at a [~~the~~] beginning of the data transfer, and the receiver assembly sending a [~~sends its~~] data-ready signal [(SCN4)] to the receiver assembly [(2)] after the data-ready signal [(SCN3)] of the transmitter assembly [(2)] has been received.

15. (Amended) A method [~~Method~~] according to claim 14, wherein said [~~characterized in that the~~] transmitter assembly [(2)] only sends its data-ready signal [(SCN3)] when a [~~the~~] complete data packet is present on said [~~this~~] assembly [(2)].

16. (Amended) A method [~~Method~~] according to claim 14 [~~or 15~~], further

comprising the step of:

setting a [~~c h a r a c t e r i z e d i n t h a t the~~] maximum size of the data packets [~~is set~~] to a predetermined value, and

only sending a data-ready signal from the receiver assembly [~~(2) only sends its data-ready signal (SCN4)~~] when there is sufficient storage space on the receiver assembly [(2)].

17. (Amended) A method [~~Method~~] according to claim 16, wherein said step of determining a maximum size determines one of [~~c h a r a c t e r i z e d i n t h a t~~] 32 bytes and [,] 64 bytes and [,] 96 bytes and [~~or~~] 128 bytes [~~are determined~~] as the maximum size of the data packets.

18.(Amended) A printer [~~Printer~~] control unit for high-performance printers, comprising: [~~having~~]

an I/O-module,

at least one [~~or more~~] raster modules [~~(9)~~] and

a serializer module [~~(10)~~], [~~whereby the modules (8 to 10) each have~~]

a processor for each of said modules, and [~~c h a r a c t e r i z e d i n t h a t the modules are connected to~~]

a parallel databus [~~according to one of the claims 1 to 12~~].

A new abstract is added as follows:

ABSTRACT OF THE DISCLOSURE

A parallel databus assembly and method includes modules connected to parallel signal lines, each of the modules having a databus driver being in immediate connection with the signal lines and a controller connected to the databus driver. At least some of the parallel signal lines are data lines for transmitting data or control lines for controlling the data transmission of the data via the data lines. A clock generator for generating a bus frequency with which the signals transmitted in the signal lines are clocked is connected to the databus drivers. The databus drivers are fashioned such that the signals to be transmitted from and to the data lines and the control lines are accepted during a clock pulse prescribed by the clock generator and are emitted during a following clock pulse.

9/PRTS

09/807352
JC02 Rec'd PCT/PTO 12 APR 2001

DATABUS AND METHOD FOR THE COMMUNICATION OF TWO ASSEMBLIES BY MEANS OF SUCH A DATABUS

The invention relates to a parallel databus and to a method for the communication of two assemblies by means of such a databus. The invention particularly refers to a parallel databus, which is suitable for a multiprocessor architecture. Given such a multiprocessor architecture, a plurality of processor systems basically having equal rights can communicate with one another via the databus.

The multibus II (multibus is a registered trademark of the Intel Corporation) represents such a databus. The multibus II is a synchronized bus defined in IEEE Standard for a High-Performance Synchronous 32-bit bus: MULTIBUS II, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, NY 10017, USA, 1988. In order to make it more simple, the "MULTIBUS II" is referred to as "multibus" in the following.

The hardware realization of such a multibus consists of a backplan [sic], in which the signal lines of the bus are arranged and which are provided with approximately 20 cable connectors, whereby an assembly can be respectively connected thereto. Figure 3 schematically shows two assemblies 2 that are connected via a multibus 1. Each assembly 2 has a databus driver 3 that is immediately connected to the signal lines of the multibus 1 and has a controller 4 that is connected to the databus driver 3. The controller 4, in turn, is connected to the electronic physical units of the assembly 2. These electronic physical units can have a processor or merely represent a passive digital circuit.

The controller 4, corresponding to the protocol of the multibus 1, logically edits the data generated by the electronic physical units and forwards them to the databus driver 3. The databus driver 3 converts the data into electrical data signals that are

appropriate for the multibus and applies the electrical data signals to the signal lines. Data signals coming from the multibus 1, in a reversed way, are accepted by the databus driver, which forwards the data to the controller 4. The controller 4 correspondingly edits the data for the processing by the electronic physical units.

The databus drivers are transparent electronic physical units, i.e., the respective corresponding input side and output side of the databus driver assumes the same logical value. Since the databus drivers are transparently fashioned, an active connection between two assemblies 2 is logically through-switched from the controller 4 of the one assembly 2 to the controller 4 of the other assembly 2.

The signal propagation time between the two controllers 4 limits the maximum transmission frequency or, respectively, bus frequency. Corresponding to the aforementioned IEEE standard, the bus frequency is 10 MHz. A transmission rate of 40 byte/s is obtained by such a bus frequency.

A study “20MHZ MULTIBUS II PARALLEL SYSTEM BUS INVESTIGATION, TAUFIK MA, INTEL CORPORATION, 8 APRIL, 1991" planned to operate the multibus with a bus frequency of 20 MHz. For this purpose, extensive adaptations and modifications have been proposed in order to optimize the individual runtimes between the controllers and databus drivers or, respectively, between the databus drivers connected via the multibus. The aim of this study is to operate a multibus having 10 assemblies at a maximum and 20 MHz, and to operate a multibus having 20 assemblies at a maximum and 16 MHz. The signal propagation time between the controllers of two assemblies would have to be reduced to 50 ns or less. The result of this study is that such an “accelerated” multibus is theoretically possible, however, there would be a considerable developing outlay until its actual realization. Younger data busses, such as the PCI bus, do not have bus drivers in order to obtain faster access to the signal lines of the databus and therefore obtain a higher throughput. These data busses, however, are limited with respect to the number of assemblies to

be connected at a maximum, which is normally clearly smaller than 10, and its physical expanse is limited to 10 cm, for example. On the other hand, a multibus can be up to 50 cm long and can connect 20 assemblies to one another, whereby a plurality of assemblies can represent processor systems of equal rights.

"Färber, G., Bussysteme, R. Oldenbourg Verlag, Munich 1987 (2. edition)" describes functions and structures of bus systems on the pages 16- 19. On page 19, image 13 shows a handshake transmission. The article "Packer, Stephen et. al., Message Passing Supports Multiple Processor Design, Computer Design" of 15 June 1984, pages 117 - 120, 122 and 124 describes measures for improving the communication in the multibus II.

An object of the invention is to create a parallel databus, which allows a high data throughput and which still has the advantages of the known multibus, such as the high number of connectable assemblies, the large physical expanse and the possibility of a multiprocessor architecture. Another object of the present invention is to create a method for the communication between two assemblies, which are respectively provided with a processor, by such a databus.

This object is achieved by a parallel databus having the features of claim 1 and by a method having the features of claim 13. Advantageous embodiments of the invention are cited in the subclaims.

The inventive parallel databus has a plurality of parallel signal lines, whereby a plurality of assemblies can be connected thereto, whereby each assembly has a databus driver being in immediate connection with the signal lines and a controller that is connected to the databus driver, whereby a sub-number of the signal lines represent data lines for transmitting the data and control lines for controlling the data transmission of the data via the data lines, and is provided with a clock generator for generating a predetermined bus frequency, with which the signals transmitted in the

signal lines are clocked. This parallel databus is characterized in that the databus drivers are connected to the clock generator and in that the databus drivers are fashioned such that the signals to be transmitted from and to the data lines and control lines are accepted during a clock pulse prescribed by a clock generator and are emitted during a subsequent clock pulse. In this way, a signal to be transmitted from one assembly to another assembly, during a first clock pulse, is transmitted from the controller of the transmitter assembly to the databus driver of the transmitter assembly, is transmitted from the databus driver of the transmitter assembly via the signal lines to the databus driver of the receiving assembly during a second clock pulse and is transmitted from the databus driver of the receiving assembly to the controller of the receiving assembly during a third clock pulse. During a clock pulse, the signals are merely transmitted between a controller and a databus driver of an assembly or between two databus drivers of two different assemblies, so that the physical signal path is kept short. These short signal paths allow correspondingly short signal propagation times, which can be kept less than or equal to 25 ns, for example, so that a bus clock pulse, for example, of 40 MHz is possible. As a result thereof, the data throughput is considerably increased, although all advantages known about the multibus are kept.

The method cited in claim 13 for the communication of two assemblies, which respectively have a processor, by such a parallel databus is characterized in that the data packets are merely acknowledged by one single handshake when the data packets are exchanged between the two assemblies. Therefore, the handshake can be fashioned so as to be distributed over a number of clock pulses, whereas the data packet can be transmitted with maximum transmission speed (= one data word per clock pulse).

The invention is subsequently explained on the basis of an exemplary embodiment shown in the drawing. Schematically shown are:

Figure 1 a printer control unit with an inventive databus,

Figure 2 two assemblies connected via an inventive databus,

Figure 3 two assemblies connected via a known databus, and

Figure 4 - 12 timing diagrams for explaining the signal transmission by means of the inventive databus.

The inventive parallel databus 5 is subsequently explained in greater detail on the basis of a printer control unit 6 for high-performance printers 7. Such a printer control unit 6 has an I/O module 8, one or more raster modules 9 and a serializer module 10. The individual modules 8 to 10 are connected to one another via the databus 5. The raster modules 9 and the serializer module 10 are connected to one another via a further pixel bus 11.

The I/O module 8 receives the bits of printing information from a computer means, which can be a large computer system or also a computer network. The I/O module 8 forwards the printing information to the raster modules 9 and the serializer module 10, whereby the raster modules 9 receive the bits of printing format information and convert them into a printing format data stream that can be processed by the high-performance printer 7. These printing format data streams are transmitted by the raster modules 9 via the pixel bus 11 to the serializer module 10, which lines up the data streams in a predetermined sequence and forwards them to the high-performance printer 7.

The modules 8 to 10 represent assemblies 2 that are respectively connected to the databus 5, whereby each assembly has a databus driver 3 and a controller 4 (Figure 2). The databus 5 corresponds to the multibus II (multibus is a registered trademark of Intel Corp.), as it is defined in IEEE standard for "High Performance Synchronous 32-

Bit Bus: MULTIBUS II The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, NY 10017, USA, 1998", apart from the changed cited in the following description.

The hardware realization of this databus 5 is composed of a backplan [sic], in which the signal lines of the bus are arranged and which are provided with 20 to 25 cable connectors to which an assembly 2 can be respectively connected. The CSM module (Central Services Module), which executes specific start routines and which initializes the individual assemblies, represents such an assembly known from the multibus. The CSM module has a clock generator, which applies a clock signal oscillating with a predetermined bus frequency to a clock signal line of the databus 5. The bus frequency is 40 MHz in the present exemplary embodiment.

The databus drivers 3 of each assembly 2 are connected to the clock signal line 12, whereby the input and output of the databus drivers 3 can be clocked corresponding to the bus frequency or, respectively, the bus clock pulse.

In addition to the clock signal line 12, the databus 5 has further lines, such as 32 data lines for transmitting the data, control lines for controlling the data transmission, decision lines for deciding (arbitration), whichen [sic] assembly is allowed to access the databus 5, address lines and one or more lines for the supply voltage and ground. In the present exemplary embodiment, the same lines are used for transmitting the addresses and the data, so that combined address lines/data lines are present.

The databus drivers 3 are inventively connected to the clock signal line 12 and are fashioned such that the signals to be transmitted from and to the data lines and control lines are accepted during a clock pulse prescribed by a clock generator and are outputted during the following clock pulse. The part of the databus driver 3, which operates the data lines and control lines, therefore is fashioned as a non-transparent electronic component with a temporary storing function, as it can be realized by a D-

flip-flop, for example. These databus drivers 3, during a clock pulse, therefore accept the signals of the data lines and control lines coming from the databus 5, they store them and output them to the respective controller 4 during the immediately following clock or, respectively, they accept a signal coming from the controller 4 during a clock pulse, they store said signal and apply it to the databus 5 at the immediately following clock pulse. The databus drivers 3 therefore are operated in a "clocked" fashion with respect to the data lines and control lines.

For explanation purposes, it is assumed in the following that the assembly shown in Figure 2 on the left side (transmitter assembly S) initiates a data transfer to the assembly (receiver assembly E) shown on the right side. In a decision method (arbitration) known from the multibus, the transmitter assembly initially obtains the right to be allowed to access the databus 5. The transmitter assembly S therefore is also referred to as bus owner.

During a first clock pulse, the controller 4 of the transmitter assembly S transmits a data word (1 - 4 byte) to the databus driver of the transmitter assembly S. The databus driver 3 stores the data word and converts it into a signal that is suitable for the databus 5, whereby said signal is present at the signal lines of the databus 5 during the following, second clock. During the second clock pulse, these electrical data signals are accepted by the databus driver 3 of the receiver assembly 2, they are temporarily stored and are transmitted to the controller 4 of the receiving assembly E during the following, third clock pulse.

This clocked transmission of the signals divides the entire transmission path from the controller 4 of the transmitter assembly S to the controller 4 of the receiver assembly E into three sections, namely the two sections 13, 15 between the controllers 4 and the databus drivers 3 of the respective assembly 2 and the section 14 between the two databus drivers 3 of the two assemblies 2, whereby said section extends across the signal lines of the databus 5. The signal propagation time in the individual sections is

significantly shorter than over the entire distance between two controllers, as it is necessary for transparent databus drivers in order to maintain the signal generated at the controller of the transmitter assembly until it is present at the controller of the receiver assembly. Given such a databus, the bus frequency can be significantly increased (e.g. up to 40 MHz) as a result of the inventively shortened signal propagation times of the individual sectionst [sic], which can be 25 ns, for example. Data transmission rates of up to 160 Mbyte/s can be obtained by a bus frequency of 40 MHz.

The databus drivers can be fashioned, for example, with GTL+- drivers of the company Texas Instruments, such as the module SN54GTL1655 or SN74GTL1655.

The operation of the inventive databus is subsequently explained in greater detail on the basis of the timing diagrams shown in the Figures 4 to 12.

Figure 4 shows the timing diagram for transmitting a data packet between two assemblies on the basis of the control signals SCN0, SCN2, SCN3, SNC4 and the address signals/data signals ADN. The designations of these signals are respectively provided with one of the following endings "_S", "_B"and "_E", whereby "-S" means that the signal status in the line section 13 is shown between the controller 4 and the databus driver 3 of the transmitter assembly S; the ending "_B" means that the signal status in the signal lines of the databus is shown between the databus drivers 3 of the communicating assemblies (line section 14), and the ending "_E" means that the signal in the line section 15 is shown between the databus driver 3 and the controller 4 of the receiver assembly E. The signal SCN0 shows a request phase, whereby the signal is valid in the low status (L = request phase). The signal SCN3 indicates the end of the data transmission, whereby the 0/1-transition or, respectively, low/high-transition represents the exact point in time of the end of the data transmission. SCN3 refers to the data-ready signal of the transmitter assembly, and SCN4 refers to the data-ready signal of the receiver assembly E. The two signals SCN3 and SCN4 are

the main components of a handshake between the transmitter assembly S and the receiver assembly E. The handshake has signals SCN5, SCN6 and SCN7 allocated, with which the type of error, in a way known from the multibus, is indicated by the receiver assembly E given a faulty data transmission.

The signals AND can comprise 4, 8, 16, 24 or 32 individual signals, for example, whereby the maximum number is limited by the 32 address lines/data lines of the databus 6.

The controller 4 of the transmitter assembly generates the request signal SCN0 at the beginning of a transmission of a data packet and the address of the receiver assembly is outputted (see S1 in Figure 4). The databus driver 3 of the transmitter assembly applies these signals to the signal lines of the databus 5 during the next clock pulse (S2). The request signal SCN0 and the address data reach the controller 4 of the receiver assembly E during the third clock pulse (S3); the receiver assembly E then realizes that it is to receive a data packet. The signal SCN0 is only generated during the duration of a clock pulse. During the second clock pulse S2, the controller 4 of the transmitter assembly S generates a data-ready signal SCN3, with which it indicates that the transmitter assembly S is ready for sending a data block. An identification character is simultaneously applied to the address lines/data lines for the type of data block. These signals arrive at the controller 4 of the receiver assembly E at the clock pulse S4. After the receiver assembly E has determined the data-ready signal SCN3, it checks whether it has sufficiently free storage space for accepting a data packet. The size of the data packet is fixed at the initialization of the databus and can be 32, 64, 96 or 128 byte. If there is sufficient storage space at the receiver assembly E for accepting a data packet, the controller 4 of the receiver assembly E produces a data-ready signal SCN4, which is transmitted to the controller of the transmitter assembly S during three clock pulses (S6 - S8). The receiver assembly E needs two clock pulses for accepting the data-ready signal SCN3 of the transmitter assembly S, for checking whether there is sufficient storage space and for outputting the data-ready

signal SCN4. The data-ready signal SCN4 reaches the controller 4 of the transmitter assembly S during the clock pulse S8. The handshake has been acknowledged with the receipt of this signal by the controller 4 of the transmitter assembly S.

The controller 4 of the transmitter assembly S sends data words, which normally comprise 32 bit, during the acknowledgment of the handshake, starting with the clock pulse S3 through the clock pulse S10. The controller 4 hereby sends such a data word per clock pulse, which then, with a delay of two clock pulses, arrive [sic] at the controller 4 of the receiver assembly E. The data-ready signal SCN3 of the transmitter assembly S is active until the last data word has been sent. At the same time as the last data word D7 is sent, the controller 4 of the transmitter assembly S activates the control signal SCN2 (low) in order to mark the exact end of the transmission of the data packet at its 0/1-transition (low/high transition). As can be seen from Figure 4, the SCN2 signal is synchronously transmitted in the individual transmission sections with the data word D7, so that the signal SCN2, during the clock pulse S12, is received by the receiver assembly E and also by the other assemblies connected to the signal lines of the databus 5, so that these recognize that the transmission of the data packet has been completed. A further data packet now can be transmitted between the transmitter assembly and the receiver assembly or the right to access the databus signals [sic] can be transferred to another assembly in a decision method (arbitration), which is known from the multibus.

Since each data packet, which can comprise 8 data words, for example, is inventively transmitted by one single handshake, the data words D0 - D7 can be transmitted with a maximum transmission rate (one data word per clock pulse), whereas a comparably long period of time of 7 clock pulses (S2 - S8), for example, is available for processing the handshake.

Figure 5 shows a similar diagram for transmitting a data packet that comprises only one data word D0. The acknowledgment of the handshake by the signal SCN4 is

received again at the clock pulse S8 by the controller 4 of the transmitter assembly S (see transmission process explained on the basis of Figure 4). Only after the handshake has been acknowledged and this acknowledgment signal SCN4 has been processed by the transmitter assembly S (clock pulse S10), the controller 4 of the transmitter assembly S ends the output of the signals representing the data word D0, so that it is assured that the receiver assembly E is capable of receiving the data word D0. Since only one data word D0 is transmitted here, it is the "last" data word of the packet why the control signal SCN2 indicating the end of the data transmission is also actively switched (low) during the entire time during which the data signals of the data word D0 are active.

The timing diagrams of Figure 6 and 7 show a reference read access and a reference write access. The reference accesses comprise the what is referred to as "I/O-space operation" and the "Memory Space Operations", wherein a transmitter assembly writes a data word in a register or a memory cell of another assembly or, respectively, reads is out from the memory cell.

Given the reference write access (Figure 6), the process of the control signals SCN0, SCN2, SCN3 and SCN4 exactly corresponds to the process when a data packet having one single data word is transmitted (Figure 5). The reference write access only differs from this data transfer in that an address ADR composed of two data words is initially transmitted at the address lines/data lines and the data word D0 to be transmitted is subsequently transmitted. A transmitter assembly S initiating a data transfer can read out a memory cell of a receiver assembly E with the reference read access (Figure 7). In the same way as the previous data transfers, the data transfer is started by initiating a request phase with the control signal SCN0 and by applying an address ADR, which is composed of two data words, to the address lines/data lines. After the request phase, the transmitter assembly S sets the control signals SCN2 and SCN3, whereby it thus indicates that it is ready for accepting the data from the receiver assembly E. The receiver assembly E sends a data word D0 to the transmitter assembly S and

simultaneously indicates that the data are valid in that it places the control signal SCN4. The receiver assembly E removes again the data D0 and the control signal SCN4 when it has recognized the set signals SCN2 and SCN3 of the transmitter assembly S. As soon as the transmitter assembly S has recognized the set signal SCN4, it takes over the data D0 and resets the control signals SCN2 and SCN3. The data transfer is completed.

The Figures 8 and 9 show the timing diagrams of an interconnect write access and of an interconnect read access. On the basis of these interconnect accesses, a data word can be respectively entered or, respectively, read out in a what is referred to as interconnect storage space, which is provided at each assembly. The memory cells of this storage space are addressed with an address ADR that is only composed of a data word. The interconnect accesses therefore differ from the reference accesses with respect to the size of the address, whereby the operational sequence of the control signals SC0, SCN2, SCN3, SCN4 corresponds to the transmission of the respective data word D0.

The reference access and interconnect access is of secondary importance for the bus systems, which respectively connect assemblies that are provided with a processor, since an interprocessor communication is only possible by means of the above-described data packets (messages). Given multiprocessor systems, the reference access and the interconnect access only serve the purpose of initializing and diagnosing the system or, respectively, are for the communication with periphery devices, which do not have a separate processor control.

As it is known from the multibus, the control signals ARB (5. . 0) and a bus request signal BREQ (bus request) are used for the decision (arbitration). In contrast to the address signals/data signals and the control signals, these signals are not clocked-in, since the decision lines then cannot be used in a "wired-or-modus", which is used for deciding the access rights. The databus driver therefore is transparent for these

signals. Since one clock pulse period is not sufficient as signal runtime from one controller to the other controller for transmitting the signals, the databus is provided with an additional clock pulse signal line, whereby an auxiliary clock pulse BCLK2 is applied thereto. The auxiliary clock pulse BCLK2 (20 MHz) is generated by dividing the bus frequency by two.

The signals ARB (5 . . 0) and BREQ are generated by the assemblies in the high-phase of the auxiliary clock pulse and are also queried in the high-phase of the auxiliary clock pulse BCLK2. It is thus assured that at least two clock pulse periods of the bus frequency or, respectively, of the bus clock pulse are available to the signals as signal runtime.

As shown in Figure 10, four clock pulses of the bus clock pulse or, respectively, two clock pulses of the auxiliary clock pulse are used at a minimum for the transiency of the decision signals. This period of time that is available to the decision signal for the transiency can be extended to 18 clock pulses of the bus frequency at a maximum.

Figure 11 shows a diagram showing the termination of a data transmission due to an error generated at the receiver assembly E. When the receiver assembly E notices an error, it outputs an error code at the same time as the data-ready signal SCN4 by means of the control signals SCN4 to SCN7 (not shown). The error code outputted by the signals SCN5 to SCN7 corresponds to the error code known from the multibus.

When the receiver assembly E recognizes the error code, it modifies the data transfer in that it sets the control signal SCN2. The data transfer is completed when the receiver assembly E recognizes the set control signal SCN2.

It is differentiated between what are referred to as bus errors, which generally occur in the databus system, vis-a-vis [sic] the errors generated with respect to the assemblies. Given the inventive databus, the error signal BUSERR can be only set by the

assemblies, which are a part of the respective data transfer, i.e., the transmitter assembly S and the receiver assembly E. All other assemblies connected to the databus are either in a resting state or are in the decision phase (arbitration). They must leave the arbitration phase and start again when a bus error occurs. The assemblies set the error signal BUSERR when a parity error is present at the address/data signals or control signals or when the bus protocol has been injured with respect to the control signals. This is to assure that the data arriving at an assembly are correct. Given a data transmission by means of a data packet, a writing interconnect access and a writing reference access, the error signal BUSERR is generated by the receiver assembly. Given a reading interconnect access and a reading reference access, wherein a data word is transmitted from the receiver assembly to the transmitter assembly, the error signal BUSERR is generated by the transmitter assembly.

It is important about the inventive databus that all signals of a transmitter assembly are clocked-in three times before they can be processed by the receiver assembly. This is the reason why a bus error is always recognized two or three clock pulses later, depending on whether the error has occurred on the transmitter assembly or on the databus. In this case, the receiver assembly sets the error signal BUSERR. Given a reading interconnect access or reading reference access, the error occurs on the receiver assembly, whereby the transmitter assembly sets the error signal BUSERR in the case of an error.

The chronological delay of the error signal is taken into consideration in that the controller of the transmitter assembly, which sends a data packet or carries out an access, waits for six clock pulses after the end of the data transfer until it informs the CPU of the assembly of the status of the data transfer, since an error signal can still arrive at this time.

Since only one single handshake inventively occurs for each data transfer, only this one single handshake, i.e. the data-ready signal SCN4, can be checked, so that modifications result vis-a-vis the known multibus protocol.

The parity check is carried out at the subsequently described points in time:

1. Control signals are always checked.

2. Address signals/data signals are checked during the request phase.

3. Address signals/data signals are checked during a response phase of a read access following the request phase when SCN4 is set and SCN (7 . . 5) is not set, i.e., when a correct handshake is present.

4. Address signals/data signals are checked during the response phase of a write access when SCN3 is set and when SCN (7 . .5) is not set, i.e., a correct handshake is present.

A databus protocol injury is determined when

1. SCN0 is set during a response phase,

2. SCN2 is set in the response phase and SCN3 is not set in the response phase,

3. SCN5, SCN6 or SCN7 are set during the response phase and SCN4 is not set,

4. SCN2 changes during the response phase, whereas SCN3 remains set, and
5. more data words than fixed during the configuration of the databus system are sent during a data transfer by means of data packets.

The multibus protocol treats the error described in point 5 as an error occurring at an assembly. Since the data transfer can be no longer continued after this error has occurred and since the error normally occurs after the handshake given the inventive method with one single handshake, it is evaluated as a bus error. Such a bus error with respect to a data transfer by a data packet is shown in the diagram of Figure 12. Nine data words are transmitted here instead of eight data words. Therefore, the receiver assembly sets the error signal BUSERR after the eighth data word D7. As can be seen from the timing diagram, the transmitter assembly has already completed the data transfer on its side before it is informed of the error by the receiver assembly. As a result thereof, the controller of the transmitter assembly must wait for six clock pulses until it can announce the completion of the transfer to the CPU of the assembly, since error messages of the receiver assembly can still arrive within this period of time.

The invention is not limited to a databus that is similar to the multibus but can be utilized for each parallel databus.

The invention can be summarized as follows:
It relates to a parallel databus having a plurality of parallel signal lines to which a plurality of assemblies can be connected, whereby each assembly has a databus driver being in immediate connection with the signal lines and has a controller that is connected to the databus driver. This databus is based on the known MULTIBUS II. The invention is characterized in that the databus drivers are connected to the clock generator of the databus, and that the databus drivers are fashioned such that the signals to be transmitted from and to the data and control lines are accepted during a clock pulse that is predetermined by the clock generator, and are emitted during the following clock pulse. As a result thereof, the signaling path between two assemblies connected via the databus is interrupted at the databus drivers, so that the signals cover a shorter path section during a clock pulse of the databus compared to conventional databuses with transparent databus drivers. The individual signal

propagation times are thus reduced, so that the bus frequency of the databus and therefore the data throughput can be significantly increased.

The invention also relates to a method for the communication of two assemblies that are each provided with a processor, whereby data packets are exchanged between the assemblies. The inventive method is characterized in that each data packet is acknowledged by only one single handshake. As a result of this method, data can be transmitted with a maximum transmission rate at the databus.

Reference character list

1	multibus
2	assembly
3	databus driver
4	controller
5	databus
6	printer control unit
7	high-performance printer
8	I/O-module
9	raster module
10	serializer module
11	pixel bus
12	clock pulse signal line
13	line section between the controller and the databus driver in the transmitter assembly
14	line section between the databus drivers of two communicating assemblies
15	line section between the controller and the databus driver in the receiver assembly

Patent claims

1. Parallel databus having
a plurality of parallel signal lines to which a plurality of assemblies (2) can be connected, whereby each assembly (2) has a databus driver being in immediate connection with the signal lines and has a controller (4) that is connected to the databus driver (3), whereby a sub-number of the signal lines represent data lines for transmitting the data and control lines for controlling the data transmission of the data via the data lines, and
a clock generator for generating a predetermined bus frequency, with which the signals transmitted in the signal lines are clocked,
characterized in that
the databus drivers (3) are connected to the clock generator and the databus drivers (3) are fashioned such that the signals to be transmitted from and to the data and control lines are accepted during a clock pulse prescribed by the clock generator, and are emitted during a following clock pulse.

2. Parallel databus according to claim 1,
characterized in that
the clock generator generates a bus frequency of at least 20 MHz.

3. Parallel databus according to claim 2,
characterized in that
the clock generator generates a bus frequency of approximately 40 MHz

4. Parallel databus according to one of the claims 1 to 3,
characterized in that
the databus has 32 data lines.

5. Parallel databus according to one of the claims 1 to 4,

characterized in that
a further sub-number of the signal lines are fashioned as decision lines for deciding which assembly connected to the signal lines has access priority, whereby the decision lines are connected to non-clocked open-drain outputs of the respective databus drivers (3), so that they form a wired-or logic.

6. Parallel databus according to claim 5,
characterized in that
a device for generating an auxiliary clock pulse (BCLK2 with a lower frequency than the bus frequency is provided for driving the decision lines.

7. Parallel databus according to claim 6,
characterized in that
the device for generating an auxiliary clock pulse is a frequency divider.

8. Parallel databus according to one of the claims 1 to 7,
characterized in that
the outputs of the databus driver (3) leading to the controller (4) are fashioned as low-voltage TTL outputs.

9. Parallel databus according to one of the claims 1 to 8,
characterized in that
the signal lines have a physical expanse of at least 40 cm.

10. Parallel databus according to one of the claims 1 to 8,
characterized in that
the signal lines have a physical expanse of at least 50 cm.

11. Parallel databus according to one of the claims 1 to 10,
characterized in that

a plurality of the assemblies (2) that are connected to the signal lines are respectively provided with a processor.

12. Parallel databus according to one of the claims 1 to 11,
c h a r a c t e r i z e d i n t h a t
the databus (5) is multibus-compatible.

13. Method for the communication of two assemblies (2), which are each connected to a processor, by means of a parallel databus (5) according to one of the claims 1 to 12, whereby data packets (messages) are exchanged between the two assemblies (2),
c h a r a c t e r i z e d i n t h a t
each data packet is acknowledged by only one single handshake.

14. Method according to claim 13,
c h a r a c t e r i z e d i n t h a t
a handshake respectively comprises a data-ready signal (SCN3) of the transmitter assembly (2) and a data-ready signal (SCN4) of the receiver assembly (2), whereby the data-ready signal (SCN3) of the transmitter assembly (2) is sent to the receiver assembly (2) at the beginning of the data transfer, and the receiver assembly sends its data-ready signal (SCN4) to the receiver assembly (2) after the data-ready signal (SCN3) of the transmitter assembly (2) has been received.

15. Method according to claim 14,
c h a r a c t e r i z e d i n t h a t
the transmitter assembly (2) only sends its data-ready signal (SCN3) when the complete data packet is present on this assembly (2).

16. Method according to claim 14 or 15,
c h a r a c t e r i z e d i n t h a t
the maximum size of the data packets is set to a predetermined value, and

the receiver assembly (2) only sends its data-ready signal (SCN4) when there is sufficient storage space on the receiver assembly (2).

17. Method according to claim 16,

characterized in that

32 byte, 64 byte, 96 byte or 128 byte are determined as the maximum size of the data packets.

18. Printer control unit for high-performance printers having an I/O-module, one or more raster modules (9) and a serializer module (10), whereby the modules (8 to 10) each have a processor,

characterized in that

the modules are connected to a parallel databus according to one of the claims 1 to 12.

IN THE UNITED STATES ELECTED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

"**SUBMITTAL OF DRAWING**"

APPLICANT: Robert BAUMGARTNER et al.

SERIAL NO.: EXAMINER:

FILING DATE: ART UNIT:

INTERNATIONAL APPLICATION NO.: PCT/EP99/07632

INTERNATIONAL FILING DATE: 12 October 1999

INVENTION: DATA BUS AND METHOD FOR ESTABLISHING COMMUNICATION BETWEEN TWO MODULES BY MEANS OF SUCH A DATA BUS

Hon. Assistant Commissioner for Patents
Box PCT
Washington D.C. 20231

SIR:

Enclosed herewith is a copy of the nine sheets of drawings as filed in the International application.

Respectfully submitted,

Melvin A. Robinson (reg. no. 31,870)
Schiff Hardin & Waite
Patent Department
6600 Sears Tower
Chicago, Illinois 60606
Telephone: 312-258-5790

ATTORNEY FOR APPLICANT

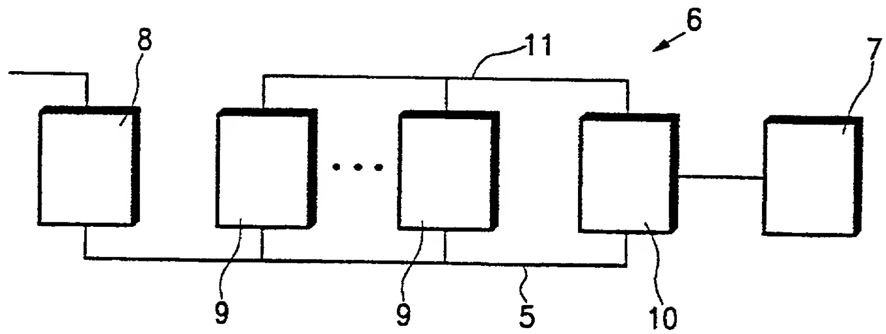

FIG. 1

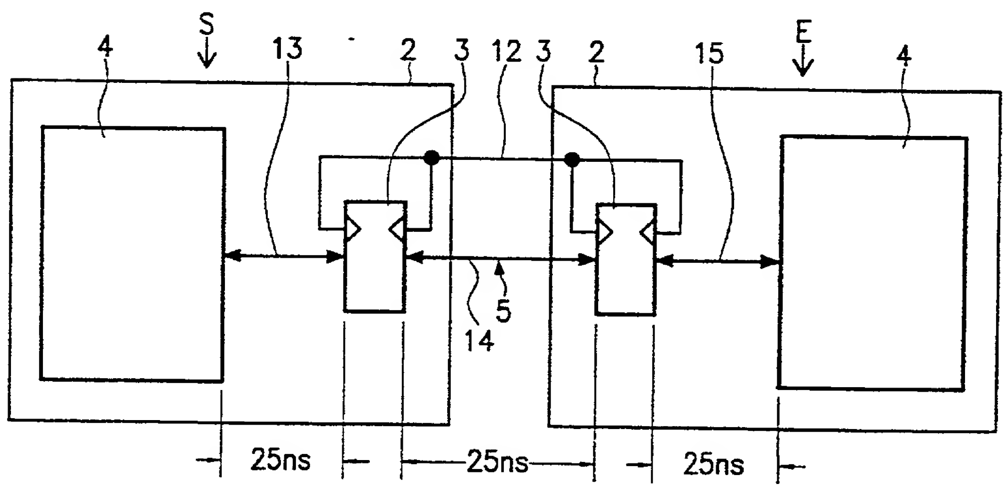

FIG. 2

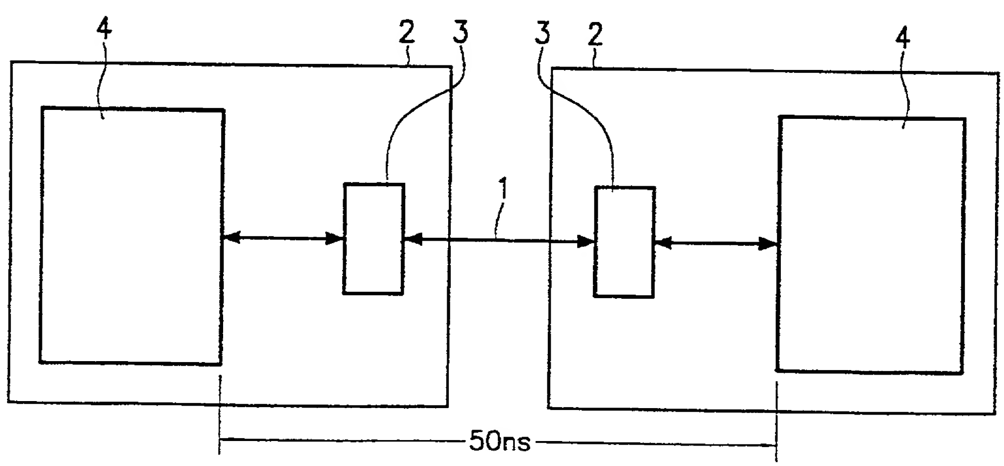

FIG. 3

09/807352

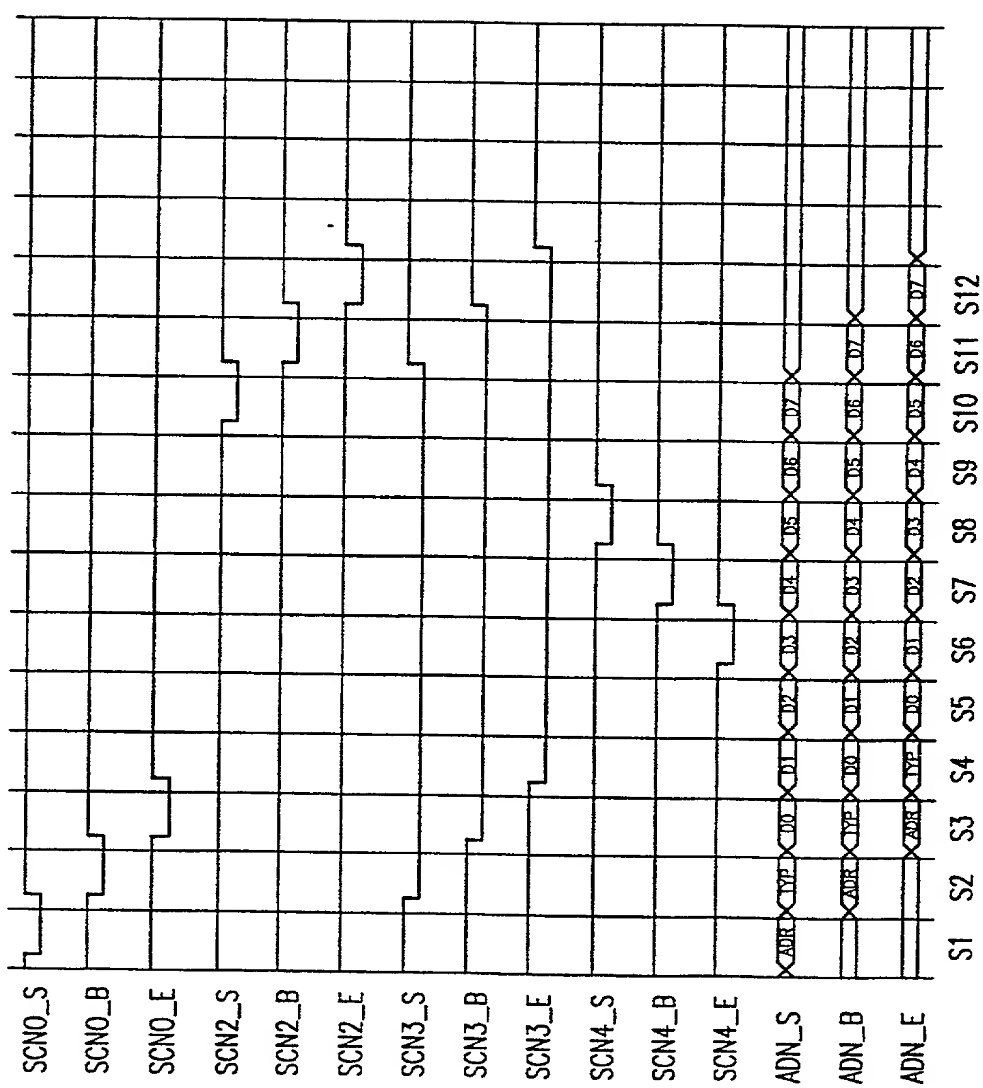

FIG. 4

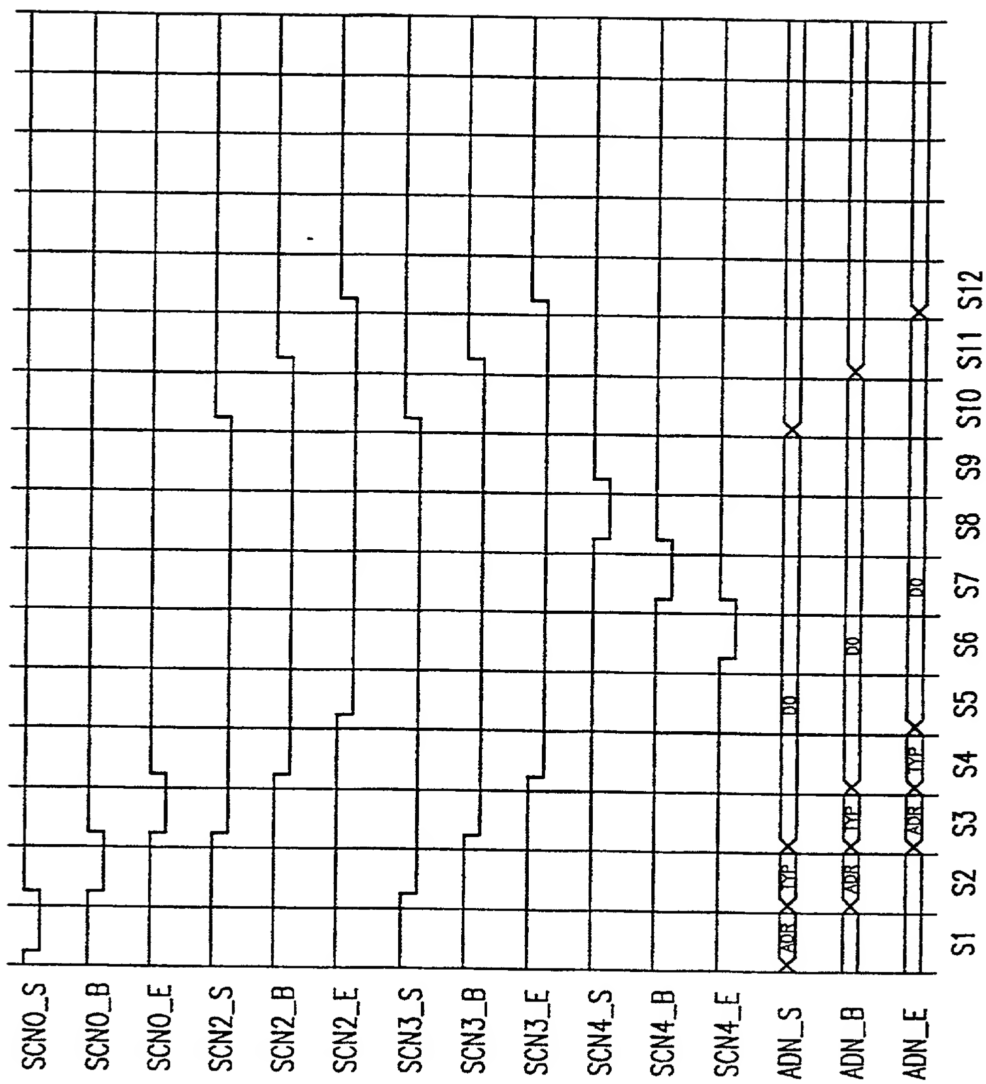
SCN0_S
SCN0_B
SCN0_E
SCN2_S
SCN2_B
SCN2_E
SCN3_S
SCN3_B
SCN3_E
SCN4_S
SCN4_B
SCN4_E
ADN_S
ADN_B
ADN_E
ADR
TYP
DO
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12

FIG.5

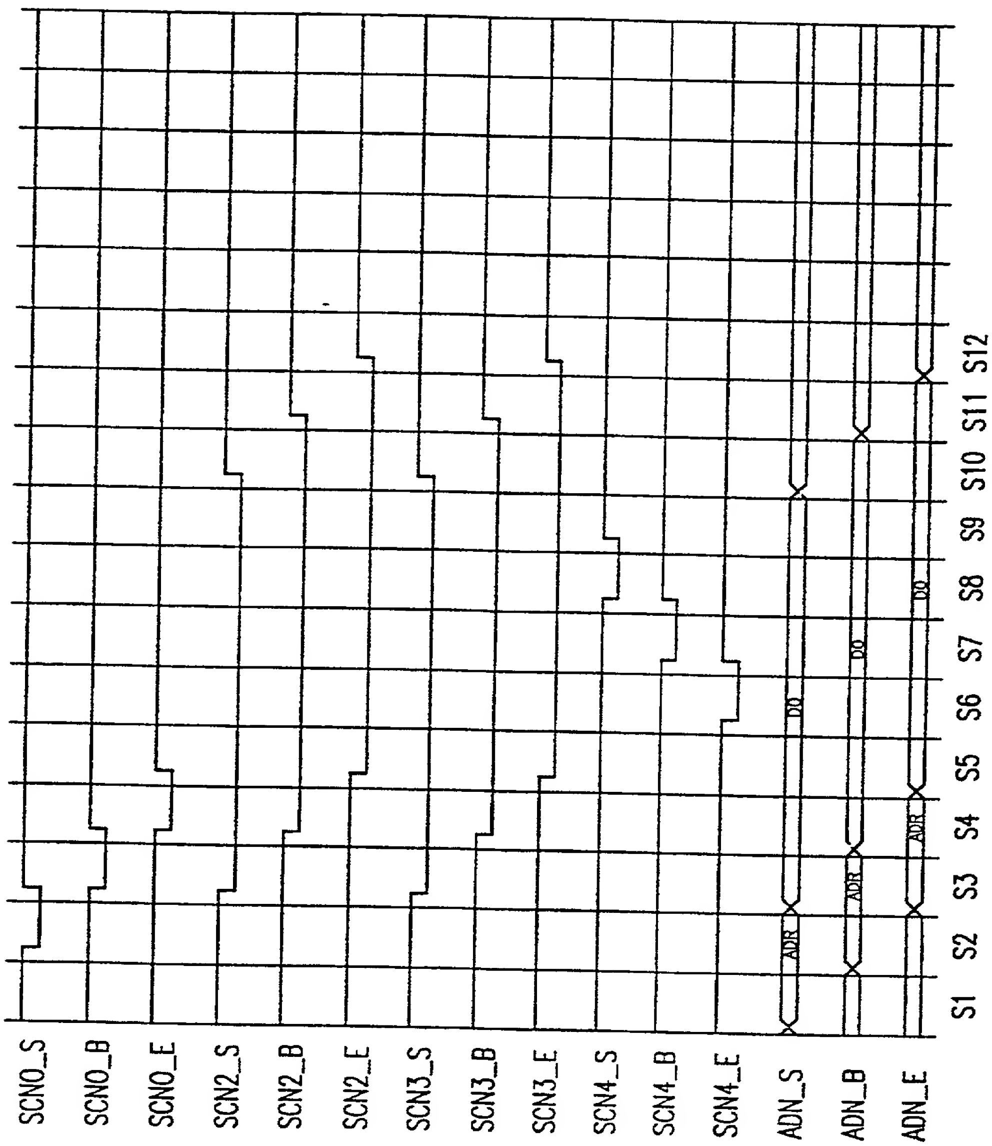
SCN0_S
SCN0_B
SCN0_E
SCN2_S
SCN2_B
SCN2_E
SCN3_S
SCN3_B
SCN3_E
SCN4_S
SCN4_B
SCN4_E
ADN_S
ADN_B
ADN_E
ADR
DO
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12

FIG.6

09/807352

5/9

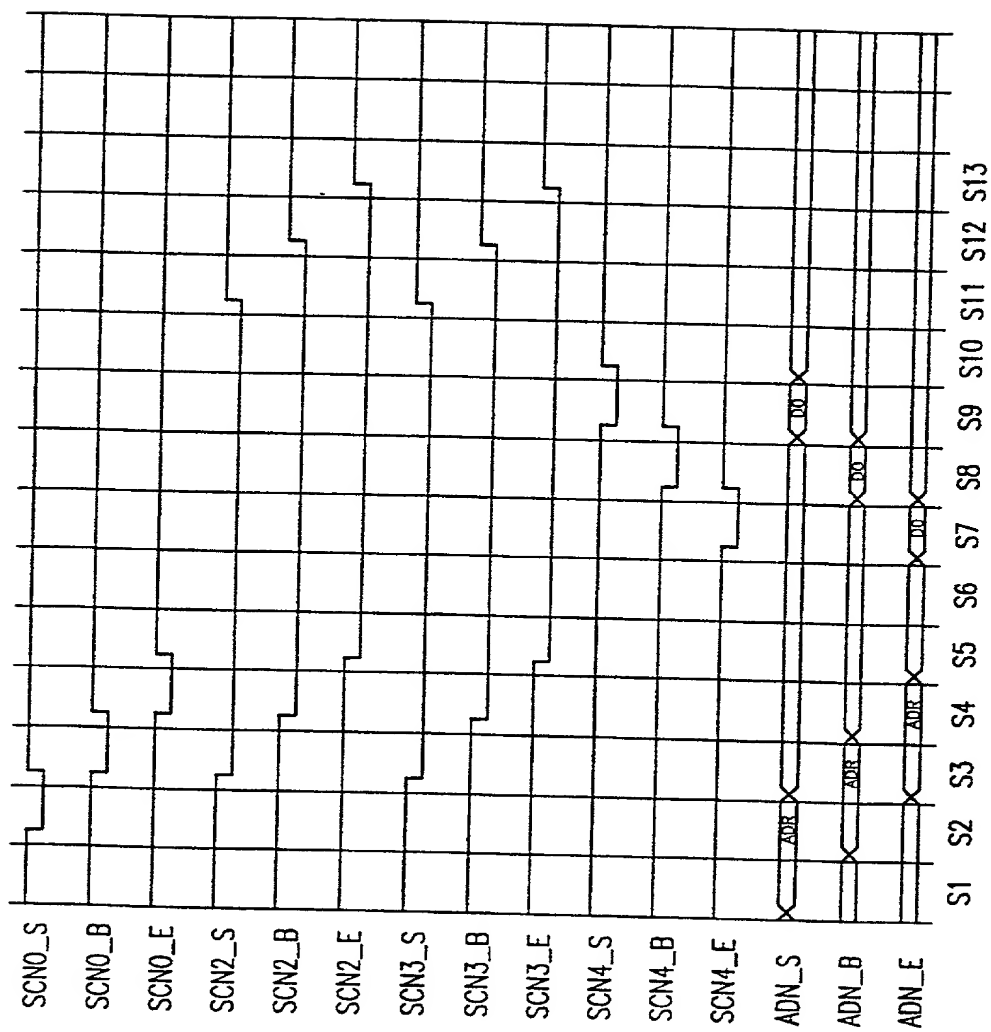

FIG.7

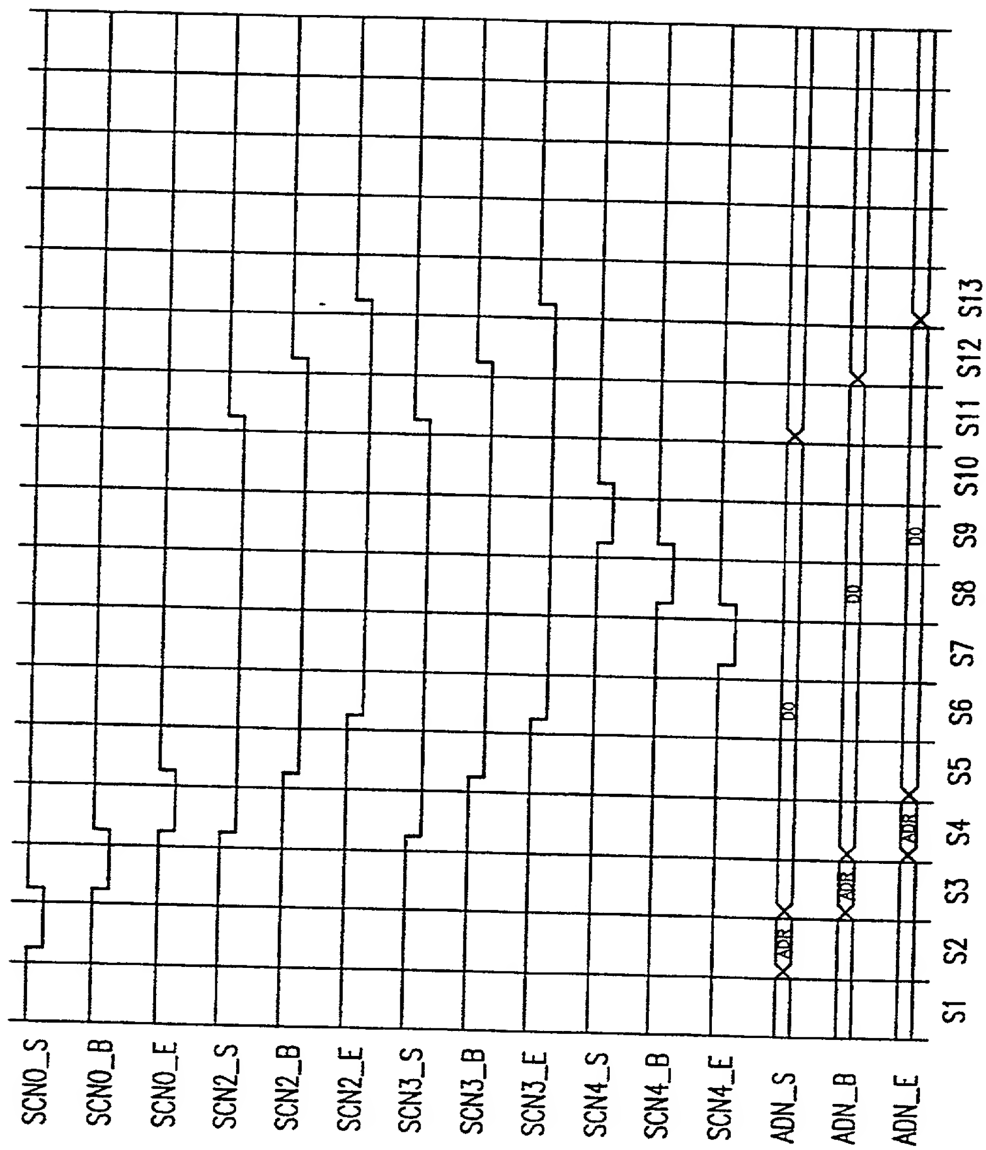
SCN0_S
SCN0_B
SCN0_E
SCN2_S
SCN2_B
SCN2_E
SCN3_S
SCN3_B
SCN3_E
SCN4_S
SCN4_B
SCN4_E
ADN_S
ADN_B
ADN_E
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
ADR
DO

FIG.8

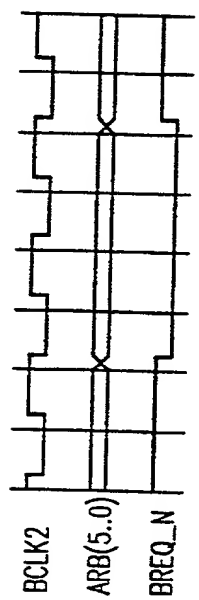
BCLK2
ARB(5..0)
BREQ_N

FIG.10

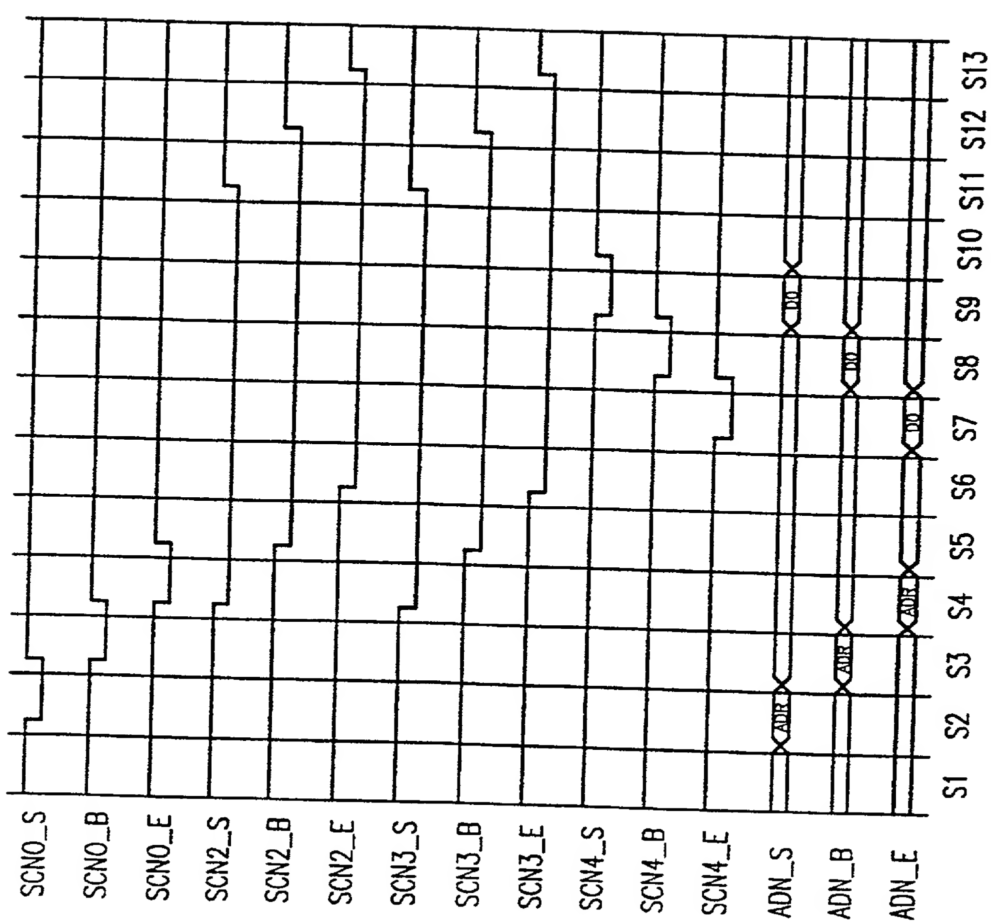
SCN0_S
SCN0_B
SCN0_E
SCN2_S
SCN2_B
SCN2_E
SCN3_S
SCN3_B
SCN3_E
SCN4_S
SCN4_B
SCN4_E
ADN_S
ADN_B
ADN_E
ADR
DO
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13

FIG.9

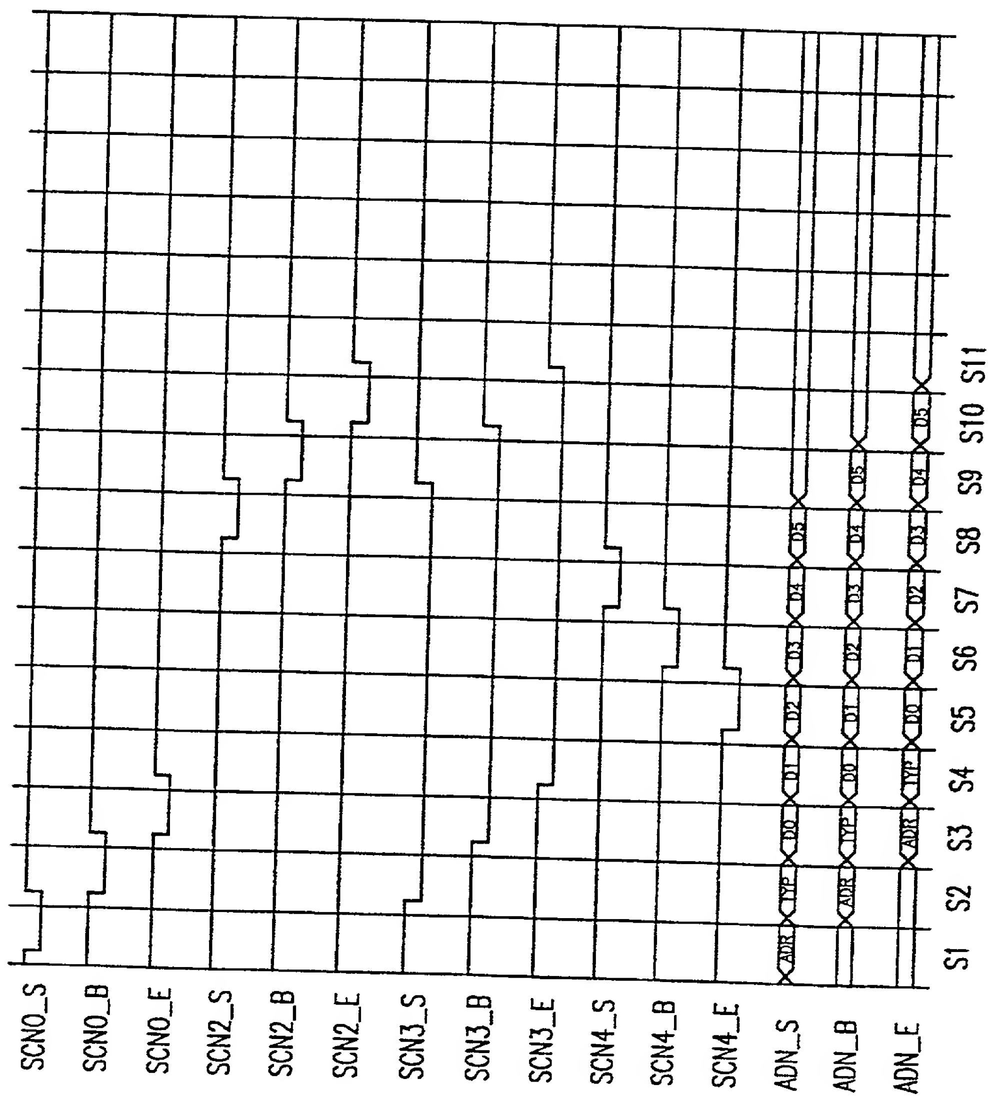
SCN0_S
SCN0_B
SCN0_E
SCN2_S
SCN2_B
SCN2_E
SCN3_S
SCN3_B
SCN3_E
SCN4_S
SCN4_B
SCN4_E
ADN_S
ADN_B
ADN_E
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
ADR
TYP
D0
D1
D2
D3
D4
D5

FIG. 11

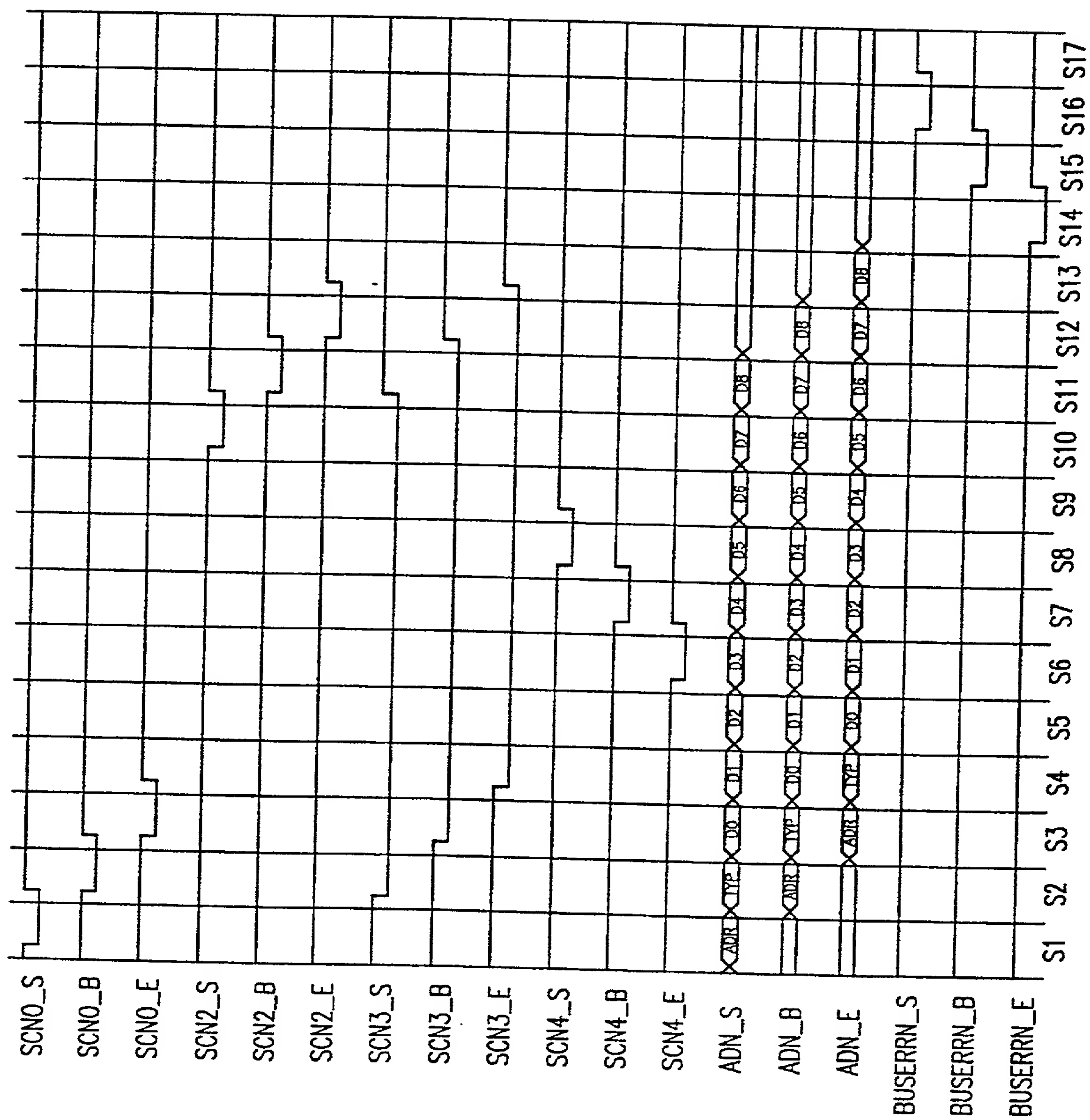
SCN0_S
SCN0_B
SCN0_E
SCN2_S
SCN2_B
SCN2_E
SCN3_S
SCN3_B
SCN3_E
SCN4_S
SCN4_B
SCN4_E
ADN_S
ADN_B
ADN_E
BUSERRN_S
BUSERRN_B
BUSERRN_E
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
ADR
TYP
D0
D1
D2
D3
D4
D5
D6
D7
D8

FIG.12

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
ERKLÄRUNG FÜR PATENTANMELDUNGEN MIT VOLLMACHT
German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für des dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

DATENBUS UND VERFAHREN ZUM KOMMNIZIEREN ZWEIER BAUGRUPPEN MITTELS EINES SOLCHEN DATENBUSSES

deren Beschreibung

(zutreffendes ankreuzen)

■ hier beigefügt ist.

■ am 12 October 1999 als PCT internationale Anmeldung PCT Anmeldungsnummer PCT/EP99/07632 eingereicht wurde und am ________ abgeändert wurde (falls tatsächlich abgeändert)

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

DATA BUS AND METHOD FOR ESTABLISHING COMMUNICATION BETWEEN TWO MODULES BY MEANS OF SUCH A DATA BUS

the specification of which

(check one)

☐ is attached hereto

■ was filed on ________ as PCT international application PCT Application No. ________ and was amended on ________ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed: